Agilent TS-5400 Functional Test System Series IIB

System Software User's Guide

Manual Part Number E8770-90033

Agilent Technologies

Notices

Documentation History

All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 (E8770-90000) (System Software User's Guide) July 1999

Edition 2 (E8770-90023)September 2000

Edition 3 (E8770-90033)September 2003

Manual Part Number

E8770-90033

Printed in USA

Agilent Technologies, Inc.
1601 California Street
Palo Alto, CA 94304 USA

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

Caution

A **Caution** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **Caution** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Safety Summary

The following general safety precautions must be observed during all phases of operation of this system. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the system. Agilent Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

General

This product is provided with a protective earth terminal. The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

WARNING: DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE
Do not operate the system in the presence of flammable gases or flames.

If the equipment in this system is used in a manner not specified by Agilent Technologies, the protection provided by the equipment may be impaired.

Cleaning Instructions

Clean the system cabinet using a soft cloth dampened in water.

WARNING: DO NOT REMOVE ANY SYSTEM COVER
Operating personnel must not remove system covers. Component replacement and internal adjustments must be made only by qualified service personnel. Equipment that appears damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Environmental Conditions

Unless otherwise noted in the specifications, this system is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 80% and at altitudes of up to 2000 meters. Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

Before applying power

Verify that all safety precautions are taken. Note the external markings described in “Safety Symbols and Regulatory Markings” on page 4.

Ground the System

To minimize shock hazard, the system chassis must have a hard-wired connection to an electrical protective earth ground. The system must also be connected to the ac power mains through a power cable that includes a protective earth conductor. The power cable ground wire must be connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective grounding will cause a potential shock hazard that could result in personal injury.

Fuses

Use only fuses with the required rated current, voltage, and specified type (normal blow, time delay). Do not use repaired fuses or short-circuited fuse holders. To do so could cause a shock or fire hazard.

Operator Safety Information

MODULE CONNECTORS AND TEST SIGNAL CABLES CONNECTED TO THEM CANNOT BE OPERATOR ACCESSIBLE: Cables and connectors are considered inaccessible if a tool (e.g., screwdriver, wrench, socket, etc.) or a key (equipment in a locked cabinet) is required to gain access to them. Additionally, the operator cannot have access to a conductive surface connected to any cable conductor (High, Low or Guard).

ASSURE THE EQUIPMENT UNDER TEST HAS ADEQUATE INSULATION BETWEEN THE CABLE CONNECTIONS AND ANY OPERATOR-ACCESSIBLE PARTS (DOORS, COVERS, PANELS, SHIELDS, CASES, CABINETS, ETC.): Verify there are multiple and sufficient protective means (rated for the voltages you are applying) to assure the operator will NOT come into contact with any energized conductor even if one of the protective means fails to work as intended. For example, the inner side of a case, cabinet, door, cover or panel can be covered with an insulating material as well as routing the test cables to the module's front panel connectors through non-conductive, flexible conduit such as that used in electrical power distribution.

Safety Symbols and Regulatory Markings

Symbols and markings on the system, in manuals and on instruments alert you to potential risks, provide information about conditions, and comply with international regulations. Table 1 defines the symbols and markings you may encounter.

Table 1 Safety Symbols and Markings

Safety symbols	
	Warning: risk of electric shock.
	Caution: refer to accompanying documents.
	Alternating current.
	Both direct and alternating current.
	Earth (ground) terminal
	Protective earth (ground) terminal
	Frame or chassis terminal
	Terminal is at earth potential. Used for measurement and control circuits designed to be operated with one terminal at earth potential.
	Switch setting indicator. ○ = Off, | = On.
	Standby (supply); units with this symbol are not completely disconnected from ac mains when this switch is off. To completely disconnect the unit from ac mains, either disconnect the power cord, or have a qualified electrician install an external switch.
Regulatory Markings	
	The CE mark is a registered trademark of the European Community.
	The CSA mark is a registered trademark of the Canadian Standards Association.
N10149	The C-tick mark is a registered trademark of the Spectrum Management Agency of Australia. This signifies compliance with the Australian EMC Framework regulations under the terms of the Radio Communications Act of 1992.
ISM 1-A	This text indicates that the product is an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4).

Service and Support

Any adjustment, maintenance, or repair of this product must be performed by qualified personnel. Contact your customer engineer through your local Agilent Technologies Service Center.

Agilent on the Web

You can find information about technical and professional services, product support, and equipment repair and service on the Web:

http://www.agilent.com

Click the link to **Test & Measurement**. Select your country from the drop-down menus. The Web page that appears next has contact information specific for your country.

Agilent by Phone

If you do not have access to the Internet, call one of the numbers in Table 2.

Table 2 Agilent Call Centers and Regional Headquarters

United States and Canada:	Test and Measurement Call Center (800) 452 4844 (toll-free in US)
Europe:	(41 22) 780 8111
Japan:	Measurement Assistance Center (81) 0426 56 7832
Latin America:	305 269 7548
Asia-Pacific:	(85 22) 599 7777

Contents

Chapter 1

System Software Overview

This chapter contains a detailed overview of the system software. Chapter contents are:

Agilent TestExec SL

The Agilent TS-5400 System uses Agilent TestExec SL and system specific software to test the Unit Under Test (UUT). Agilent TestExec SL is a test executive designed for high-volume, high-throughput functional test applications. The other system specific software provides the communications between the test executive and system instruments.

Agilent TestExec SL uses testplans (see “Testplans”) and actions (see “Actions”) to perform the tests. The actions are the building blocks from which the tests are created. The actions are called from a test which are executed in a testplan. The switching actions makes the connections from the system instruments, and/or loads on the load cards to the Unit Under Test. The switching actions are built into Agilent TestExec SL.

Both actions and switching actions are used in a testplan to run the tests. The testplan automatically closes the appropriate pin matrix and load card switches to make the connections, setup and execute the appropriate sources and detectors, and return any test results.

The switching information and the instrument types used in the system are located in the system.ust file (see “System.ust File” on page 11 for more information). This file is generated at the factory and is custom for each system. It has the necessary information for the switching actions to close the appropriate switches and for the other actions to communicate with the system instruments. Instrument data in the system.ust file can be generated and/or modified using the System Configuration Editor (see “System Configuration Editor” on page 13).

Testplans

Figure 1-1 shows a testplan and some of its components; an explanation follows the figure (see Loading a

Testplan for an explanation on how to laod a testplan).

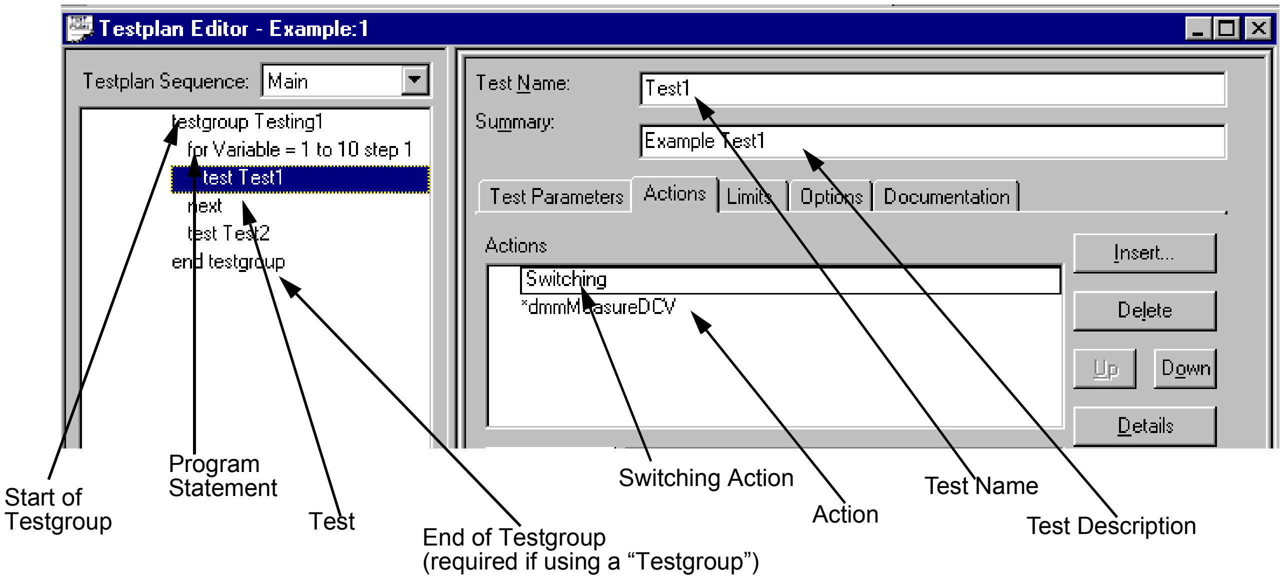


Figure 1-1. Typical Testplan Components

Testgroup

A named block of tests that can be executed in a predefined order. Each testgroup in a testplan must have a unique name; no duplicate names are allowed.

List of Statements

Test or flow control statements executed in the order shown.

Test

A named series of actions that can be executed as a group. A test can contain actions and switching actions. A test can have limit checking capabilities to determine if a test passes or fails. A test must have a unique name; no duplicate names are allowed.

Switching Action

Actions that make connections from the instrument and loads to the Unit Under Test. These actions are internal to Agilent TestExec SL. The switching information in these actions are determined by the data in the 'system.ust' file and the user-generated fixture.ust and uut.ust files.

Action

The smallest component in a test or testgroup that can be called to perform functions such as setting up an instrument, making measurements and prompting the user.

Actions

The system comes with a set of supplied actions. These actions are used for such things as configuring instruments, making measurements and prompting users for inputs. You can also generate custom actions using the Action Wizard and an application program such as Visual C/C++.

The action directory path is:

C:\Program Files\Agilent\TS-5400 System Software\actions

Actions are located in the following sub-directories:

Sub-Directory	Action Type
arb	Arbitrary Waveform Generator Actions
counter	Counter Actions
dac	D/A Converter Actions
dgn	Diagnostics Actions
digitizer	Digitizer Actions
dio	Digital I/O Actions
dmm	Digital Multimeter Actions
event	Event Detector Actions
generic	Miscellaneous Actions
mcm	Measurement Control Module Actions
power	Power Supply Actions
serial	Serial Interface Actions
SerialProtocol	Automotive Serial Protocol Actions
switch	SLU and Switching Actions

System.ust File

The 'system.ust' file is created at the factory with the appropriate module/instrument and wiring data according to the system option. The file can be edited using the System Configuration Editor (see “System Configuration Editor” on page 13) or the Topology Editor in Agilent TestExec SL.

The data in the 'system.ust' file consists of modules/instruments and instrument nodes (i.e., connections at the 32-Pin Matrix and Instrument Multiplexer Module), and other nodes (connections at the 32-Pin Matrix Modules and Load Cards) using the Aliases, Wires, and Modules designations. The Module designations are used to determine the modules/instruments installed in the system, and the Aliases and Wires are used to generate switch paths.

A typical 'system.ust' file contains Aliases, Wires and Modules.

Aliases

These are alternate names for reference nodes. The names are descriptive in nature to easily identify the node. For example, the node name for the high current output of the Agilent 34401 or E1411 Digital Multimeter is called "IsrcHi". Figure 1-2 shows typical Aliases in the 'system.ust' file.

Wires

These are names for wires that connect to or between nodes. In some cases, these names are the same aliases used for nodes. For example, the alias called "DVMIsrcHi" is often used for the wire name that connects to that node. Figure 1-3 shows typical wires in the 'system.ust' file.

Modules

These are the names of the instruments in the system. For example, the name "MCM" indicates that there is an Agilent E6171B Measurement control module in the system. The figure below shows a few typical modules in the 'system.ust' file.

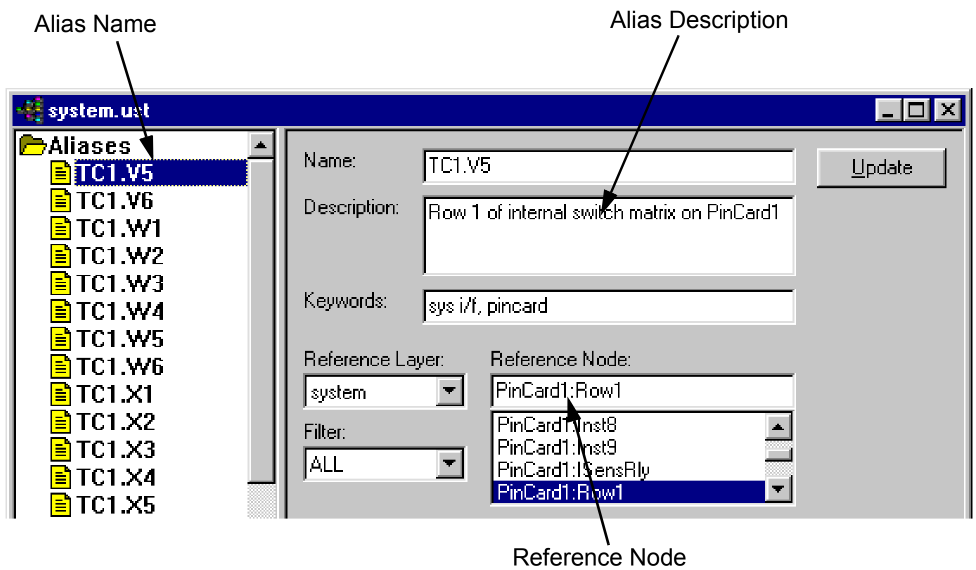


Figure 1-2. Typical Aliases

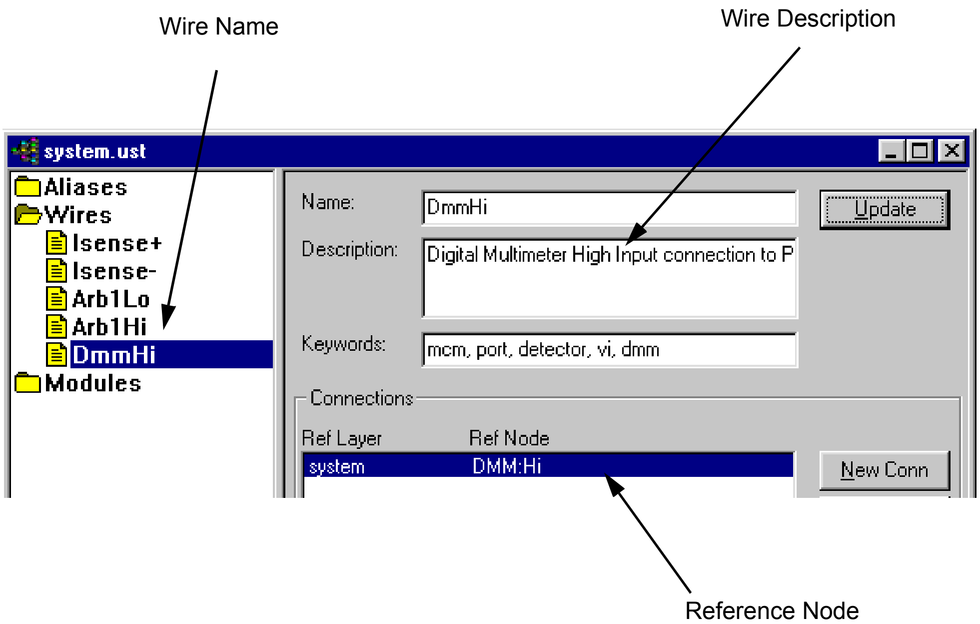


Figure 1-3. Typical Wires

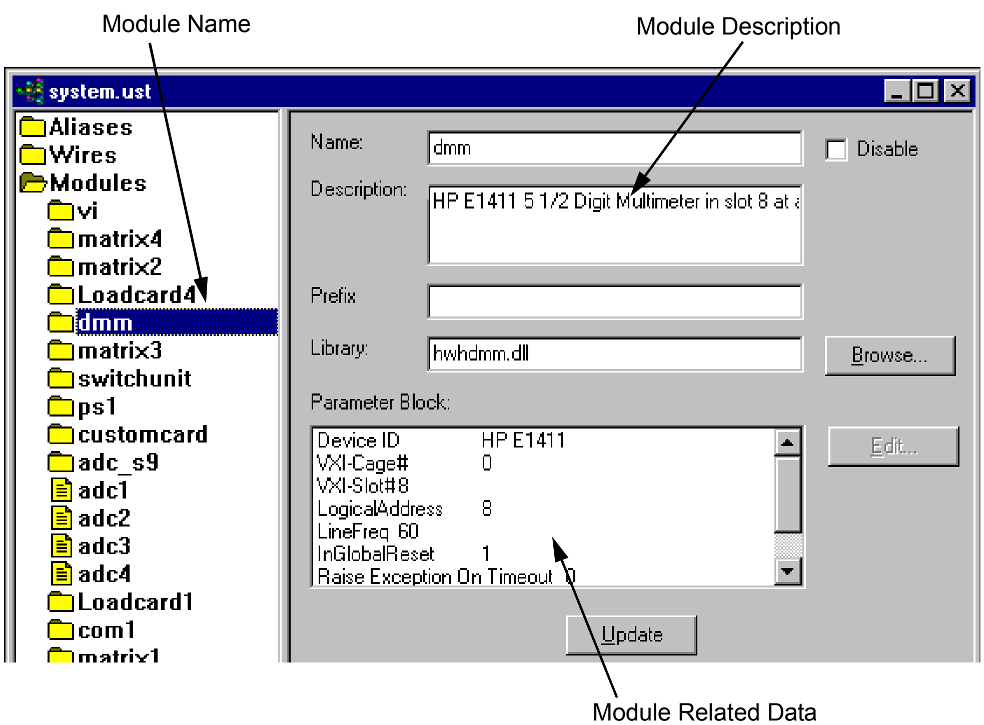


Figure 1-4. Typical Modules/Instruments

System Configuration Editor

This editor can be used to add new modules/instruments or delete old modules/instruments to/from the system.ust

file (see “System.ust File” on page 11 for information). It can add/delete any supported GPIB or VXI instruments, Pin Matrix Modules, and/or Load Cards.

The editor has the following functions:

- Shows all supported modules/instruments.
- Automatically detects all modules/instruments currently in the system
- Able to add new custom modules/instruments to the system.ust file
- Able to edit module/instrument parameters

Any of the supported and custom modules/instruments can be added to the system.ust file. Dependent on the system type and option (e.g., Test System Interface vs. Mass Interconnect), the System Configuration Editor automatically generates the appropriate wires and aliases (see “System.ust File” on page 11 for information). The wiring, etc. data is located in a spreadsheet that is specifically generated for your system type and option.

The System Configuration Editor is available from the TestExec SL Toolbar or from a shortcut in the Desktop on the system PC controller. Refer to the System Configuration Editor’s online help for more details.

Chapter 2

How to Use the System Software

Chapter Contents

This chapter lists the needed software to run the system and shows some system specific software operation. The chapter is separated as follows:

Required Computer Hardware and Software

The following is a list of the computer hardware and software needed to run the Agilent TS-5400 Test System. All necessary software was factory installed on your system:

- IBM-compatible PC (at least a Pentium) with 256 MB of RAM, 1024x768 graphics, 100 MB of free disk space (20 MB for Agilent TestExec SL software).
- Microsoft® Windows XP®.
- Agilent TestExec SL software, Version 5.1 or later.
- Agilent TS-5400 Software version 5.1 or later.

Note For more detailed information about using the Agilent TestExec SL software, refer to the software documentation

System Software Description

The Agilent TS-5400 System Series IIB uses Agilent TestExec SL and system specific software to test the Unit Under Test. Agilent TestExec SL is a test executive designed for high-volume, high-throughput functional test applications. The other system specific software provides the communications between the test executive and system instruments.

Agilent TestExec SL uses testplans and actions to perform the tests. The actions are the building blocks from which the tests are created. The actions are called from a test which are executed in a testplan. The switching actions makes the connections from the system instruments, and/or loads on the load cards to the Unit Under Test. The switching actions are built into Agilent TestExec SL.

Both actions and switching actions are used in a testplan to run the tests. The testplan automatically closes the appropriate pin matrix and load card switches to make the connections, setup and execute the appropriate sources and detectors, and return any test results.

The switching information and the instrument types used in the system are located in the `system.ust` file. This file is generated at the factory and is custom for each system. It has the necessary information for the switching actions to close the appropriate switches and for the other actions to communicate with the system instruments. Instrument data in the `system.ust` file can be generated and/or modified using the System Configuration Editor (see “Using the System Configuration Editor” in the “TS-5400 System Integrator’s Manual”.).

Selecting Agilent TestExec SL

The Agilent TestExec SL software is pre-installed on your PC controller's hard drive. Start TestExec SL from this icon in the PC desktop:

You can also run TestExec SL by clicking:

Start | Programs | Agilent TestExec SL 5.1 | TestExec SL 5.1

Loading a Testplan

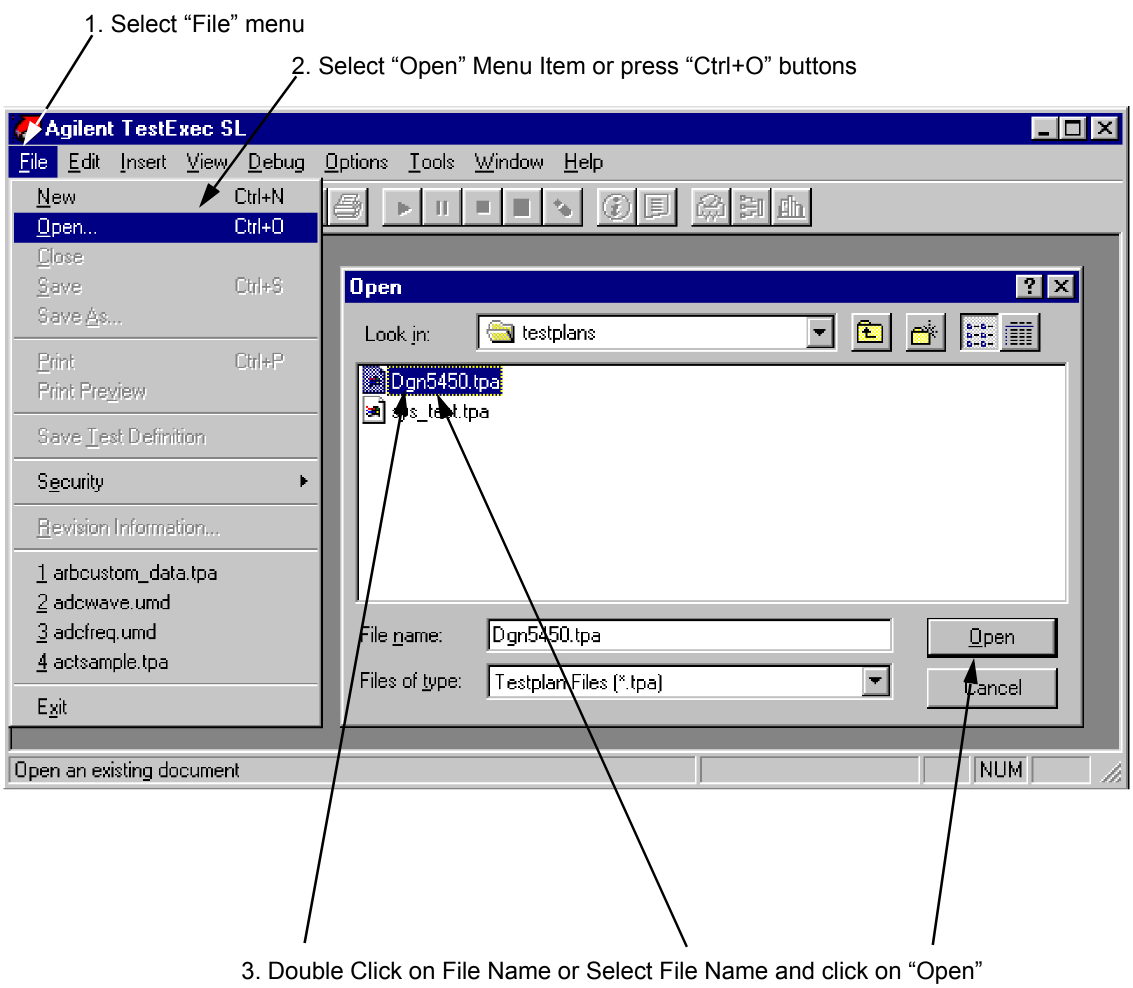


Figure 2-1. Agilent TestExec SL Main Screen and File Open Box

Creating a Testplan

Figure 2-2 shows how to create a testplan.

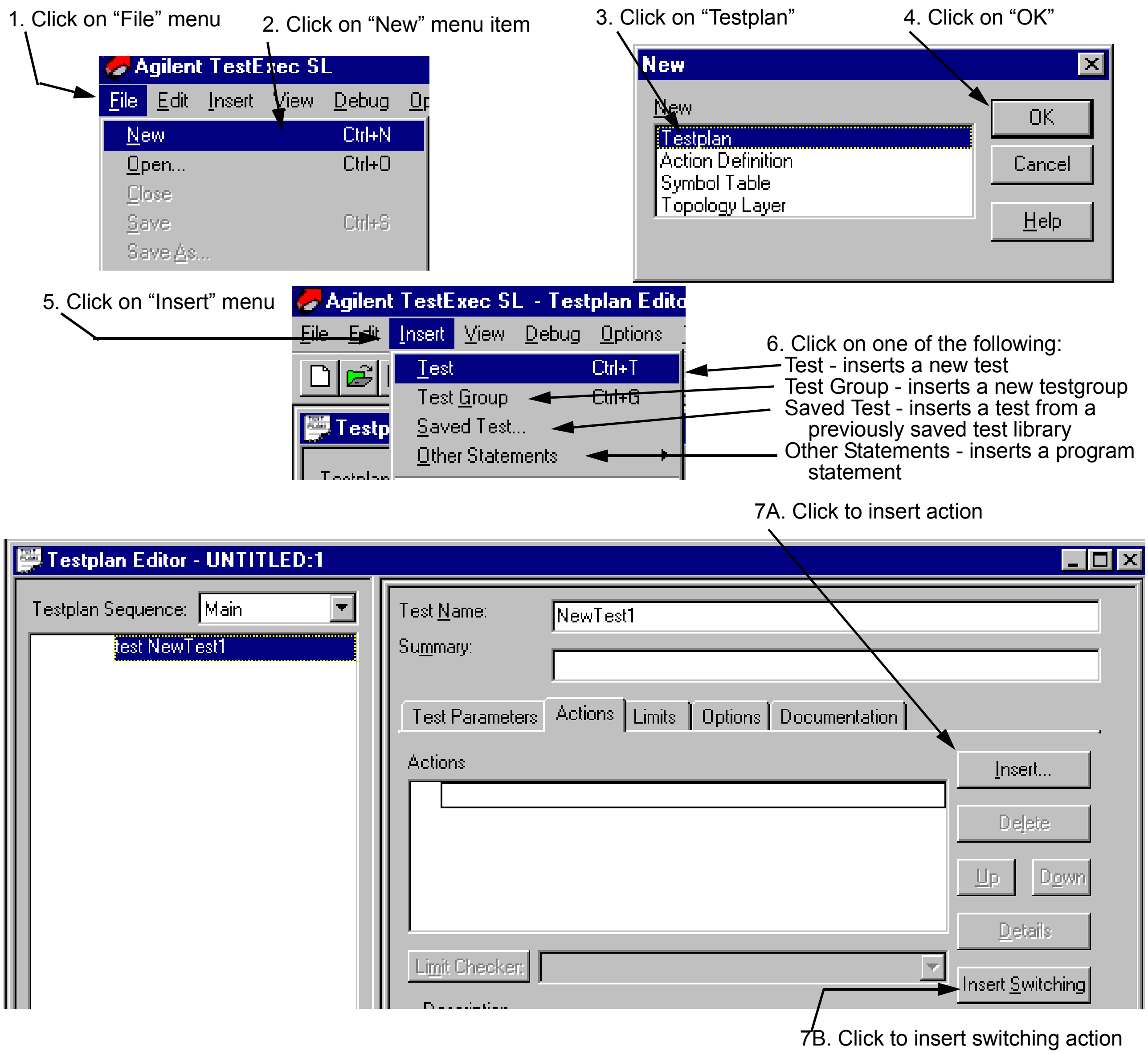


Figure 2-2. Creating a Testplan

Using TS-5400 Supplied Actions

The Agilent TS-5400 System with a set of standard actions supplied with the system. The action directory path is:

`C:\Program Files\Agilent\TS-5400 System Software\actions`

Actions are located in the following sub-directories:

Sub-Directory	Action Type
arb	Arbitrary Waveform Generator Actions
counter	Counter Actions
dac	D/A Converter Actions
dgn	Diagnostics Actions
digitizer	Digitizer Actions
dio	Digital I/O Actions
dmm	Digital Multimeter Actions
event	Event Detector Actions
generic	Miscellaneous Actions
mcm	Measurement Control Module Actions
power	Power Supply Actions
serial	Serial Interface Actions
SerialProtocol	Automotive Serial Protocol Actions
switch	SLU and Switching Actions

Note The actions are summarized in Chapter 5 of this manual. The actions are documented in detail in the TS-5400 online help which is available from the TestExec SL Help menu.

Standard Action Types

The standard actions are generally organized around a specific instrument or module in the system. The actions are also separated into the types discussed below.

High-Level Actions

These are actions that usually perform complete tests, such as setting up a source and then making a measurement. These actions normally, but not always, use one or more of the Low-Level Actions.

Low-Level Actions

These actions perform specific configuration or measurement function on a particular instrument. The low-level actions give you more flexibility in configuring instruments and making the measurements than do the high-level actions. Low-Level actions are organized by instrument type and function. For example, an action that is to set up triggering for a voltmeter is a low-level action.

Example Testplans

Example testplans to show how to use the actions are in the following directory:

C:\Program Files\Agilent\TS-5400 System Software\testplan\examples

Adding an Action to a Testplan

Do the following:

1. Open Agilent TestExec SL using the procedure in “Selecting Agilent TestExec SL” on page 17.
2. Either create a new or open an old testplan, using the procedure in “Loading a Testplan” on page 17 or “Creating a Testplan” on page 18, respectively.
3. If using a new testplan, do the procedure in Figure 2-2 on page 18 to add a test. If using an old testplan, select the test into which you wish to add an action and continue with step 4.
4. Add an action to the test, as shown in Figure 2-3 and Figure 2-4.

   There are two different actions that can be added, a regular action and a switching action. Figure 2-3 shows how to add a regular action and Figure 2-4 shows how to add a switching action.

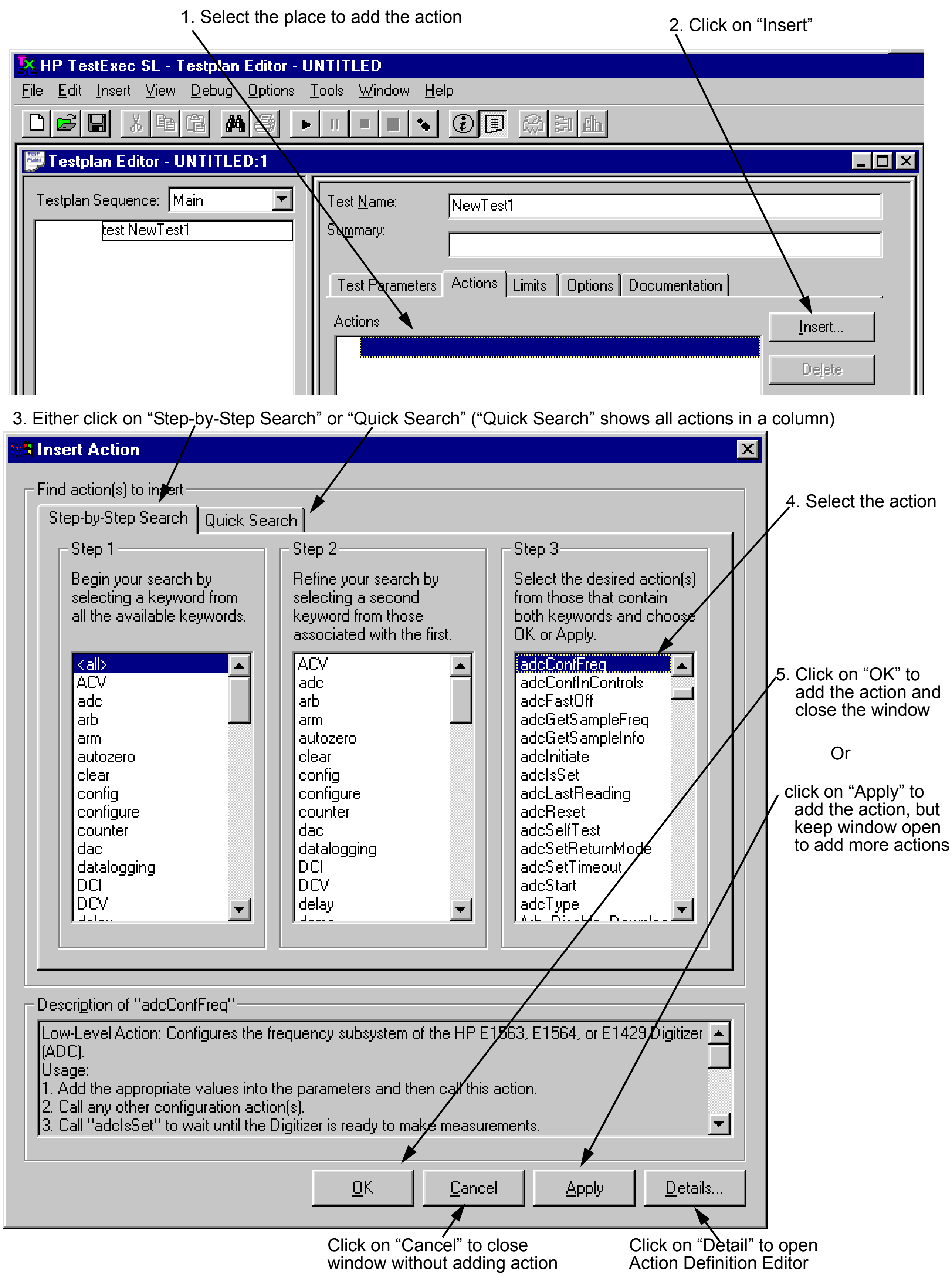


Figure 2-3. Adding an Action

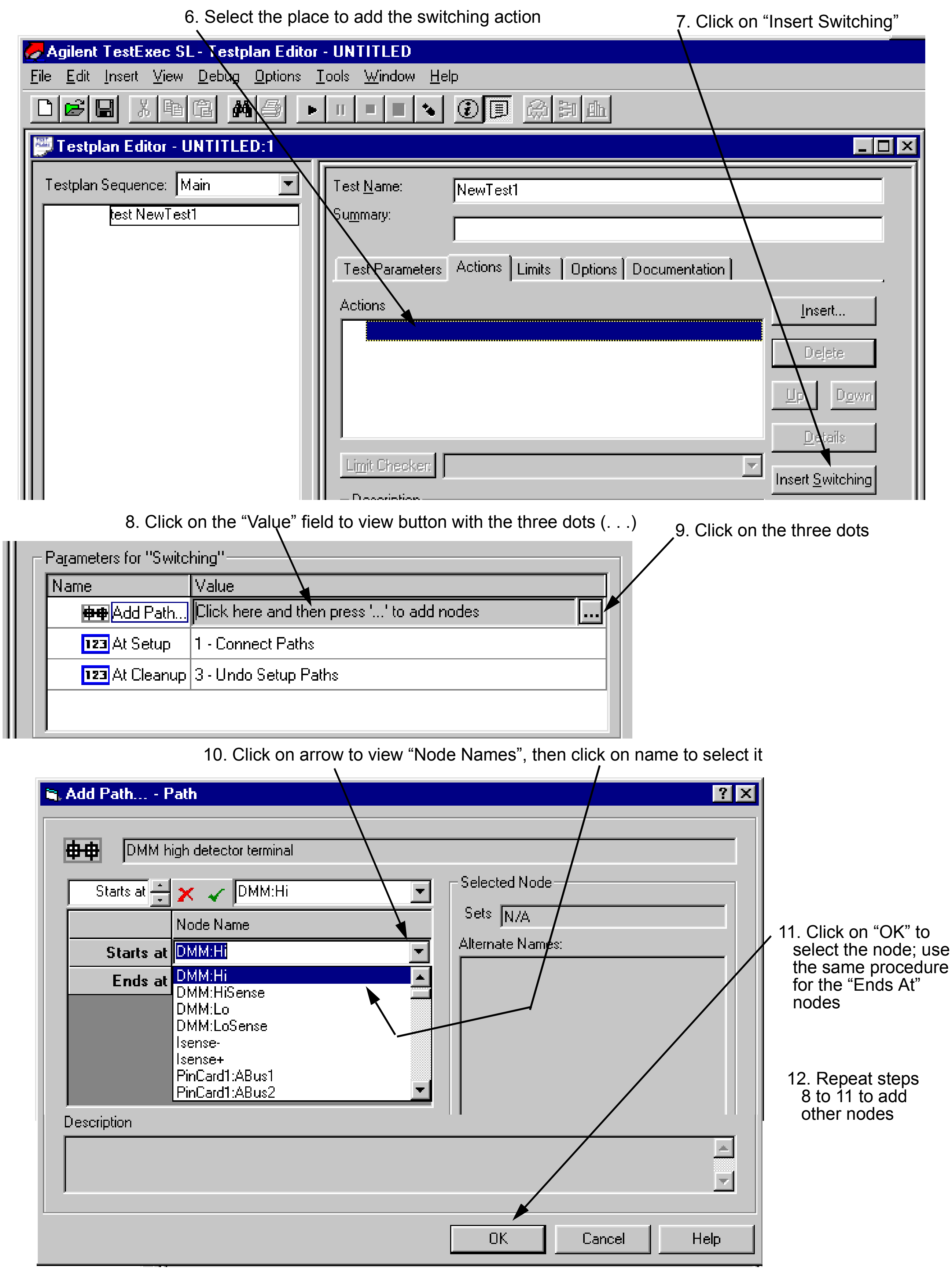


Figure 2-4. Adding a Switching Action

Using Software Debug Features

This section describes specific software features, unique to the Agilent TestExec SL version 2.0 and later, that you will find helpful in creating and debugging your tests and testplans. To enable Agilent TestExec SL, use the procedure in “Selecting Agilent TestExec SL” on page 17, if not enabled.

Looping

The loop constructs are `FOR / NEXT` sequencer statements. For example, a testplan developer would stop a testplan (if one is running), add `FOR` and `NEXT` lines to the testplan and then restart the testplan from the beginning. Refer to the Agilent TestExec SL documentation for additional information.

Figure 2-5 shows how to select and insert the looping statements into a testplan.

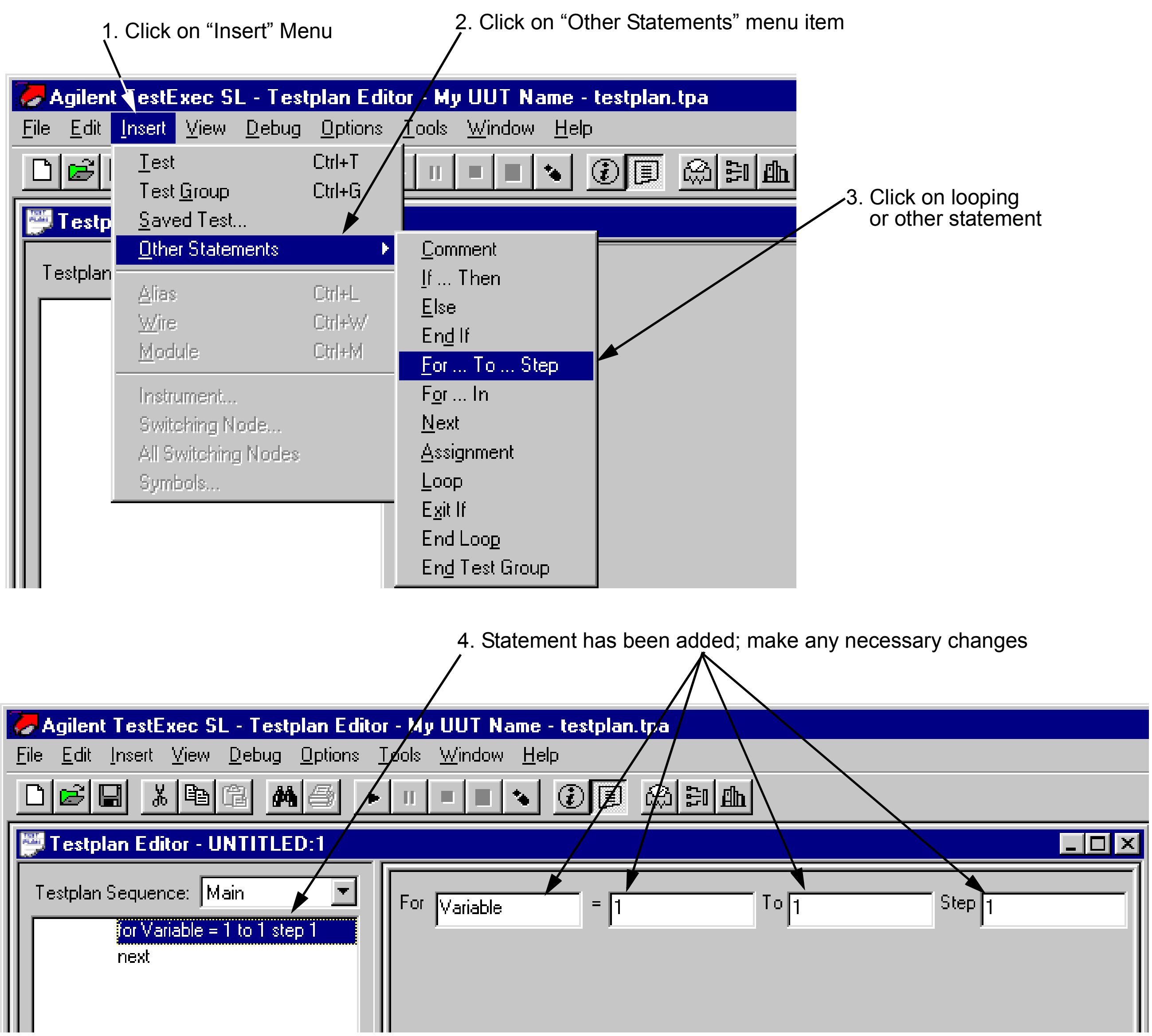


Figure 2-5. Selecting Looping and Other Statements

Single Stepping

Agilent TestExec SL adds the ability to single step the execution of the operations that make up a test. When stepping through the actions of a test, the system pauses at entry point calls for each action routine. Action stepping can be performed any time testplan execution is paused (as a result of operator pause or test breakpoint). The action step which is currently paused is listed in the trace window. Refer to the Agilent TestExec SL documentation for additional information.

Figure 2-6 shows how to select the single stepping

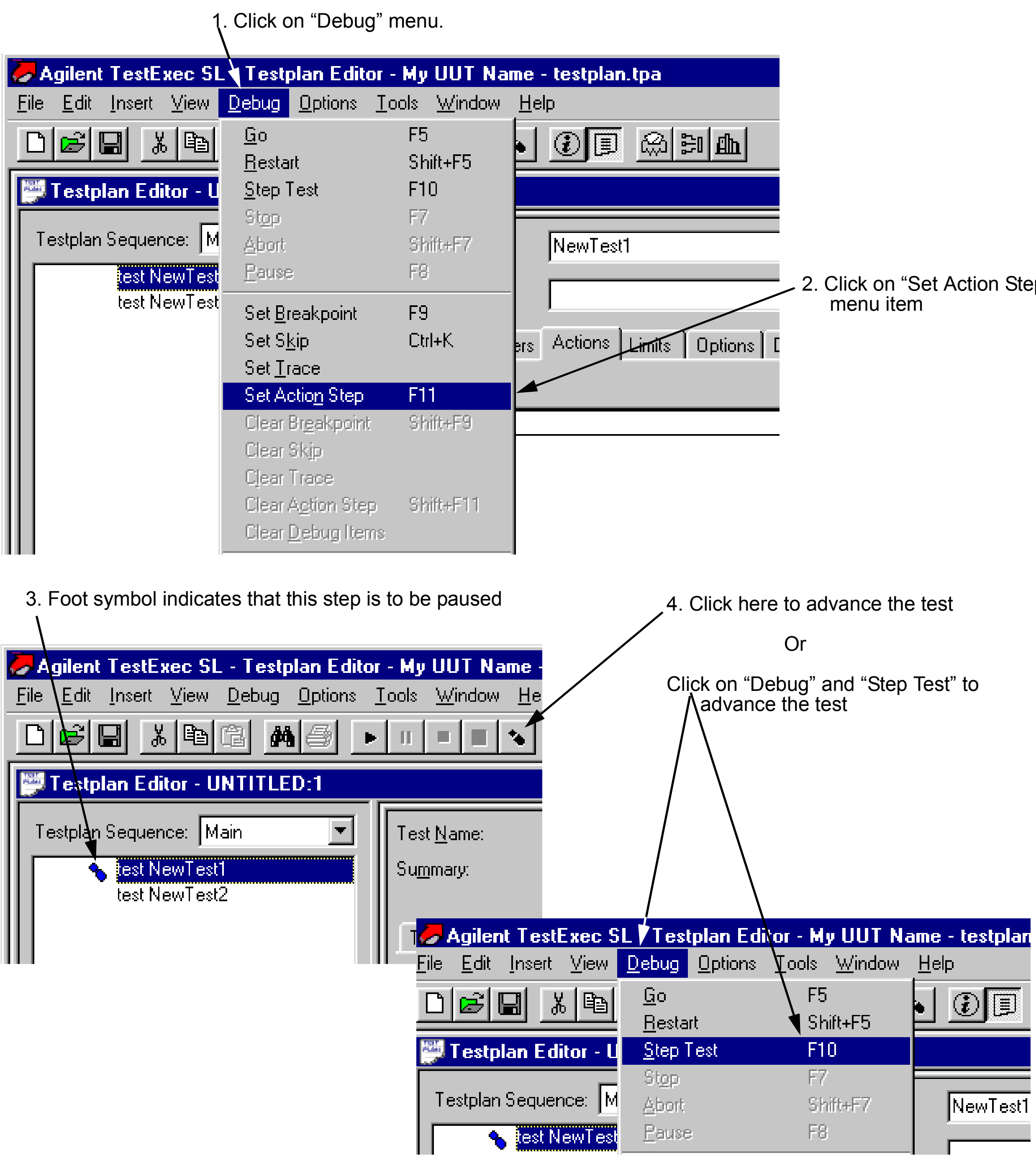


Figure 2-6. Selecting Action Stepping

Action Debug Messages

Agilent TestExec SL allows action routines to send strings to the same debug window that the system uses for the test trace output. Use the following API:

- UtaTrace (MessageString, MessageID)

  -- MessageString is a string that will be sent to the trace window.
  -- MessageID is an optional string that allows the messages to be group (as identified by MessageID). Note: currently, this parameter does not provide any functionality.

Watch Window

Watch windows provide the ability to select 'objects' to dynamically monitor as a testplan executes. Select an item to watch from a list, the item is added to the watch window list and is updated each time the testplan pauses. Items can be added or deleted at any time (as long as the testplan is not running).

The watch window can monitor 4 types of objects:

- Instruments
- Switching Node
- All Switching Nodes
- Symbols

The following shows some typical example of the window.

Watch Window Menu Selections

Watch Window Menu Selections shows how to select one of the four Watch Window objects. The object to be selected, as shown in the figure, is the 'instrument' object. (The selection of the other objects is the same.)

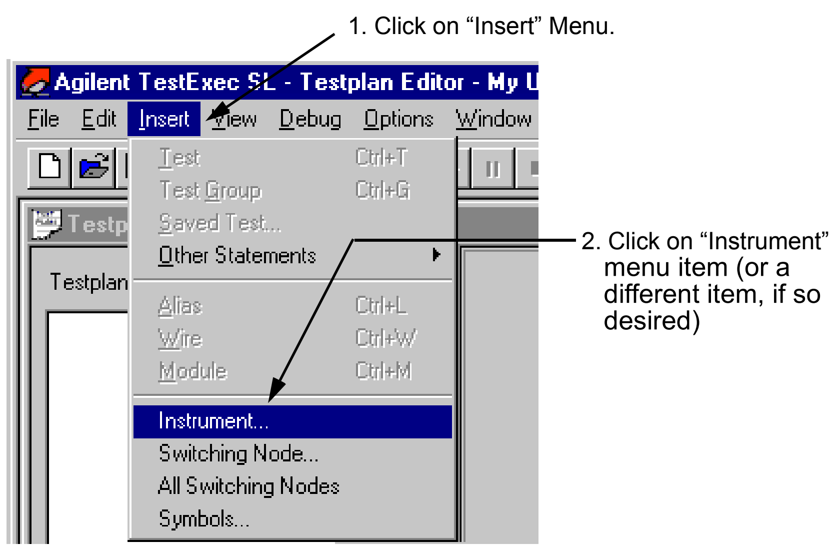


Figure 2-7. Watch Window Menu Selections

Watch Window Instrument Status

Watch windows are available for many of the instruments in an Agilent TS-5400 system. The instruments that can be monitored are:

- Agilent E1328A or E1418A Digital to Analog Converter Module (DAC)
- Agilent E1333A Counter Module
- Agilent 34401/E1411 Digit Multimeter Module (DMM)
- VXI Technology E1563A Analog to Digital Converter Module (ADC)
- Agilent E6171B Measurement Control Module (MCM)
- Agilent E6174A 32-Channel Event Detector Module
- Agilent 6642A, 6643A, 6652A, 6653A, 6673A Power Supplies

The displayed instrument states match the parameters of the instrument handler routines. For example the Agilent 34401/E1411 Digital Multimeter has the following routines which affect the state of the module:

- dmmConfFunction (dmm, Func, Range, Aperture);
- dmmConfCal (dmm, Linefreq, Autozero);
- dmmConfTrigIn (dmm, Trigselect, Trigcount, Delay);
- dmmConfSample (dmm, Sampsrc, Count, Period);

Figure 2-8 shows the menu box to select the instrument to be watched. The figure shows how to select the Digital Multimeter (dmm).

To add an instrument to the Watch Window, select the instrument type and press “OK”, as shown in Figure 2-8. The Watch Window is then displayed, as shown in Figure 2-9.

Figure 2-9 also shows the expanded tree view that shows the status of the ‘dmm’ parameters. The instruments display their state in a tree view. This allows a logical grouping of states and reduces the amount of data displayed by placing less important states in lower levels of the tree.

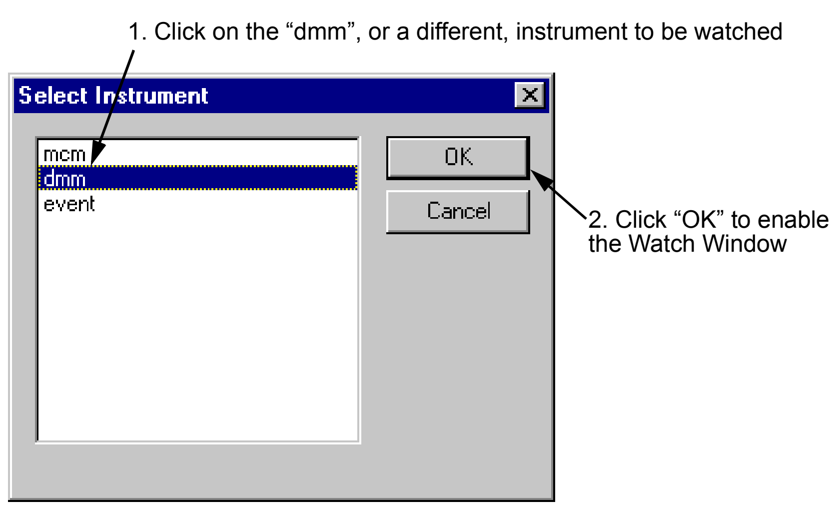


Figure 2-8. Selecting an Instrument to be Watched

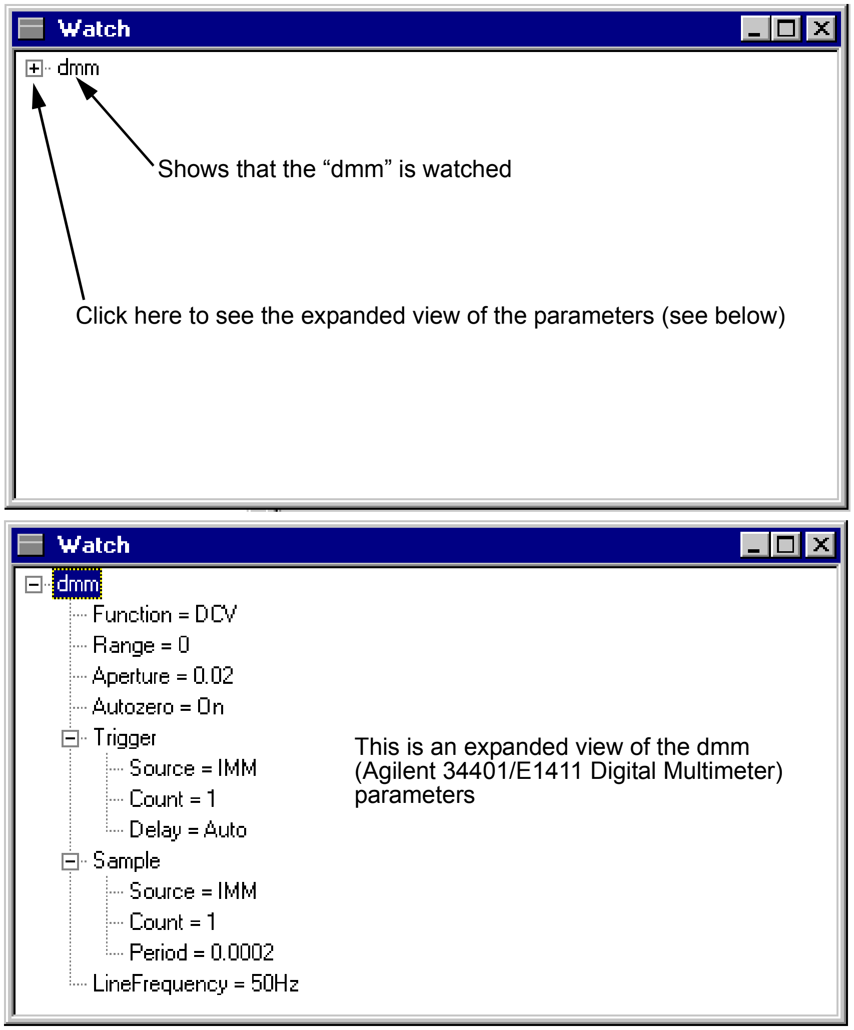


Figure 2-9. Agilent E1411B Digital Multimeter Watch Window with Parameters

Instrument Debug Front Panels

Agilent TestExec SL provides a series of debug instrument front panels for the Agilent TS-5400 system instruments. The debug panels provide two main features:

- The ability to interactively view the current Unit Under test (UUT) state
- The ability to interactively view the current state of points within the system.

Debug Panel Types

The following debug panels are supplied with the system, dependent on the modules/instrument currently installed in the system.

Debug Panel Features

All of the debug panels provide the following features:

- Changes on the instrument front panel are not made until either:

  -- you initiate a measurement (for detectors) or
  -- you press the “Apply” button (for sources)

  The instrument state memory is modified in the instrument handler at this time (since this is when the instrument handler is called; see See “Instrument Handlers” on page 49 for information about handlers).

- If the instrument is connected in the Agilent TS-5400 system through switching, the front panel provides a section which allows you to define the switching path. Either select a UUT pin to connect to directly or define a custom switch path.

  Connecting to the UUT pin is the easiest way to select a switch path; you select an ABus pin and the UUT pin. If the instrument has both the high and low terminals connected to the switch matrix, then both pins need to be defined. The low terminal is connected to UUTCommon by default and does not need to be selected, if you use the default. The UUT option is available if you have defined a UUT switching configuration file. In addition to the UUT pins, the system automatically adds the option to measure an ABus pin only. This will allows you to inject a signal on the front of the system and measure it using the instrumentation within the system.

- The state of the system is saved upon entering a pause state and is restored when entering the run state. This allows the panels to be used

without affecting the testplan.

Selecting the Debug Panel

Debug Front Panel Selection and Select an Instrument from the Debug Front Panel Selection show how to enable the debug instrument front panels.

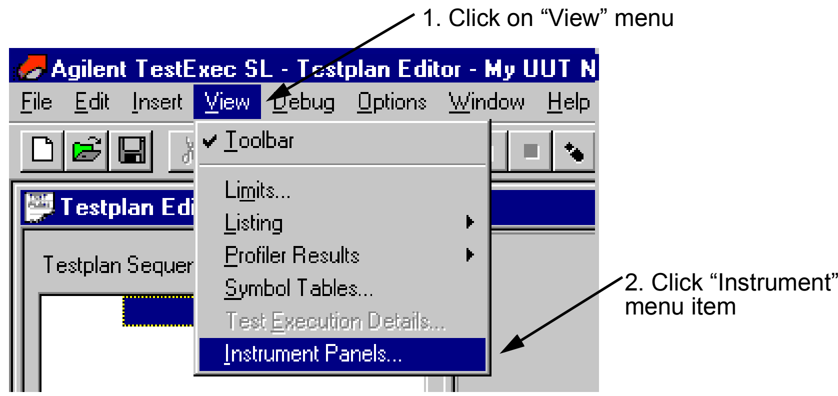


Figure 2-10. Debug Front Panel Selection

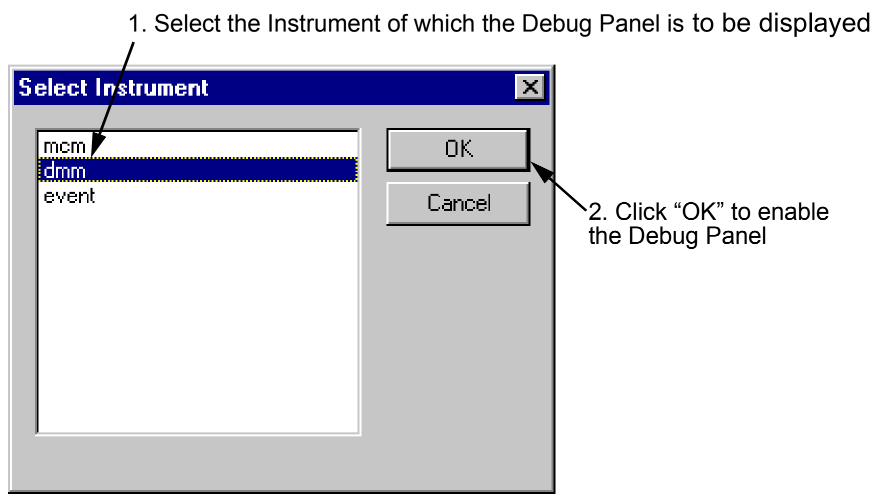


Figure 2-11. Select an Instrument from the Debug Front Panel Selection

The following sections show and explain the different debug instrument panels.

Agilent E1411B Digital Multimeter

An instrument box which connects to the Agilent E6171 Measurement Control Module, such as the Agilent E1411 DMM, usually has two different dialog boxes which depend on the type of path selected (as shown in Agilent E1411 DMM Debug Front Panel (showing Path Selection)A and Agilent E1411 DMM Debug Front Panel (showing Path Selection)B). This allows you to either connect to the pin on the UUT (Agilent E1411 DMM Debug Front Panel (showing Path Selection)A) or select any point within the system by defining a custom switching path (Agilent E1411 DMM Debug Front Panel (showing Path Selection)B). The custom switching is applied on top of the current switching state, so to undo a switching path you need to define a 'disconnect' path.

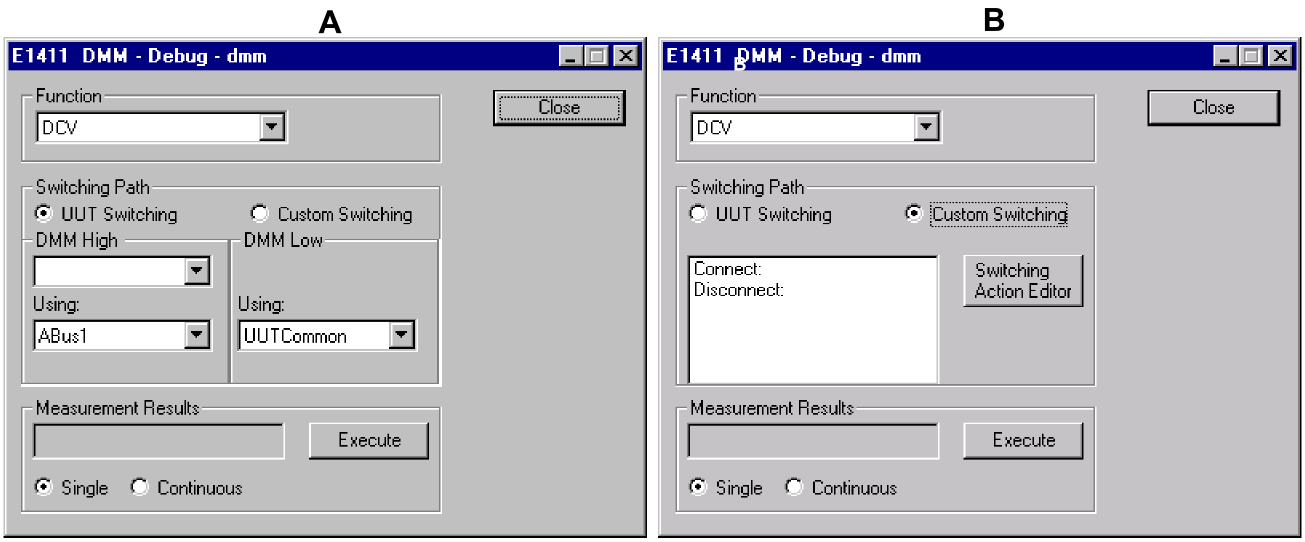


Figure 2-12. Agilent E1411 DMM Debug Front Panel (showing Path Selection)

The debug panel boxes and buttons do the following:

Function:
Choose ACV, DCV, Ohms, Ohms - Offset Compensated
The measurement is made in auto range and medium resolution.

Switching Path:
The debug panel supports the selection of the switching path. The panel switches both high and low connections of voltage sense and the high and low connection of the current terminals. The sense connections are connected in parallel with the high and low terminals. You specify the high or low terminal and the system automatically connects the associated sense terminal.

Measurement Results:
Press the "Single" button to initiate a single measurement; press the "Continuous" button to initiate a series of measurements. The continuous button is modified to a "Stop" button once pressed. When the dialog box is brought up, the measurement results box is blank.

Switching Action Editor:
This button enables the editor to select the switching paths for the

Digital Multimeter. See “Fast Connection Selection” on page 41 for a description on how to use the editor.

Execute:
This button starts a measurement.

Close:
This button closes the debug panel.

System Interface

The setting of the Digital Multimeter function uses the following actions:
dmmMeasureDCV, dmmMeasureACV, dmmMeasureOhms

The above actions also return the results of a measurement.

Agilent E1328A & E1418A Digital to Analog Converter

Agilent E1328A and E1418A DAC Debug Front Panel shows the dialog box for controlling the Agilent E1328/E1418 Digital-to-Analog Converter (DAC) Module.

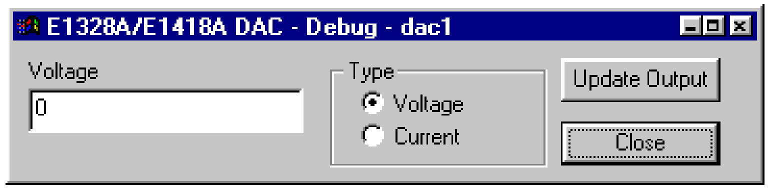


Figure 2-13. Agilent E1328A and E1418A DAC Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage or Current:
You have the option to specify the output voltage or current for each channel of the DAC. The 'Type' selection sets the voltage or current mode and defines contents of the input field.

Dependent on the type selected, enter the voltage value/current value into the Voltage/Current field.

(Note the E1328A cannot programmatically select the voltage or current option.)

Update Output:
This button changes the output setting.

Close:
This button closes the debug panel.

System Interface

The setting of the DAC uses the following action routine:
dac16i_setup

Agilent E1333A Counter

Agilent E1333 Counter Debug Front Panel shows the dialog box for Agilent E1333A Counter. Only the UUT connection screen is shown. The custom path selection is the same used by the Digital Multimeter (see Figure 2-12 on page 30).

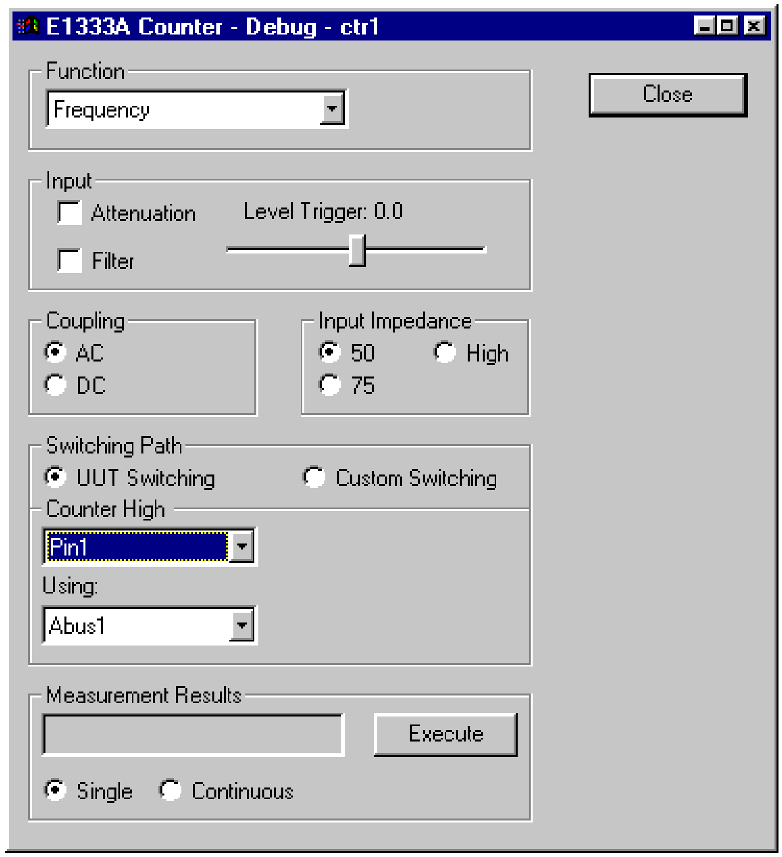


Figure 2-14. Agilent E1333 Counter Debug Front Panel

The debug panel boxes and buttons do the following:

Function:
Choose Frequency, Period, Positive Pulse Width, Negative Pulse Width, Totalize

Input, Coupling, Input Impedance:
Allows you to set input functions. The setting of each function is sent before a measurement is made.

Switching Path:
The counter panel supports the selection of a switching path. The counter switches only the high connections of the output. The low connections are automatically connected to system ground.

Measurement Results:
Select the "Single" button to initiate a single measurement or the "Continuous" button to initiate a series of measurements. The continuous button is changed to a "Stop" button, once pressed.

Note that when the dialog box is brought up, the measurement results box is blank.

Execute:
This button starts a measurement.

Close:
This button closes the debug panel.

System Interface

The setting of the counter function will use the following action routines:
ctrMeasureFrequency, ctrMeasurePeriod, ctrMeasurePulseWidth, ctrMeasureTimeInterval, ctrMeasureTotalize

The measurement uses the level trigger specified in the dialog box (using the slider control), a default range and default resolution.

The input section uses the following action routine:
ctrMeasureInControls

VXI Technology E1563A Digitizer

Figure 2-15 shows the debug front panel for the VXI Technology E1563A Digitizer. Only the UUT connection screen is shown. The custom path selection is the same as for the Agilent E1411B Digital Multimeter (see “Agilent E1411B Digital Multimeter” on page 30).

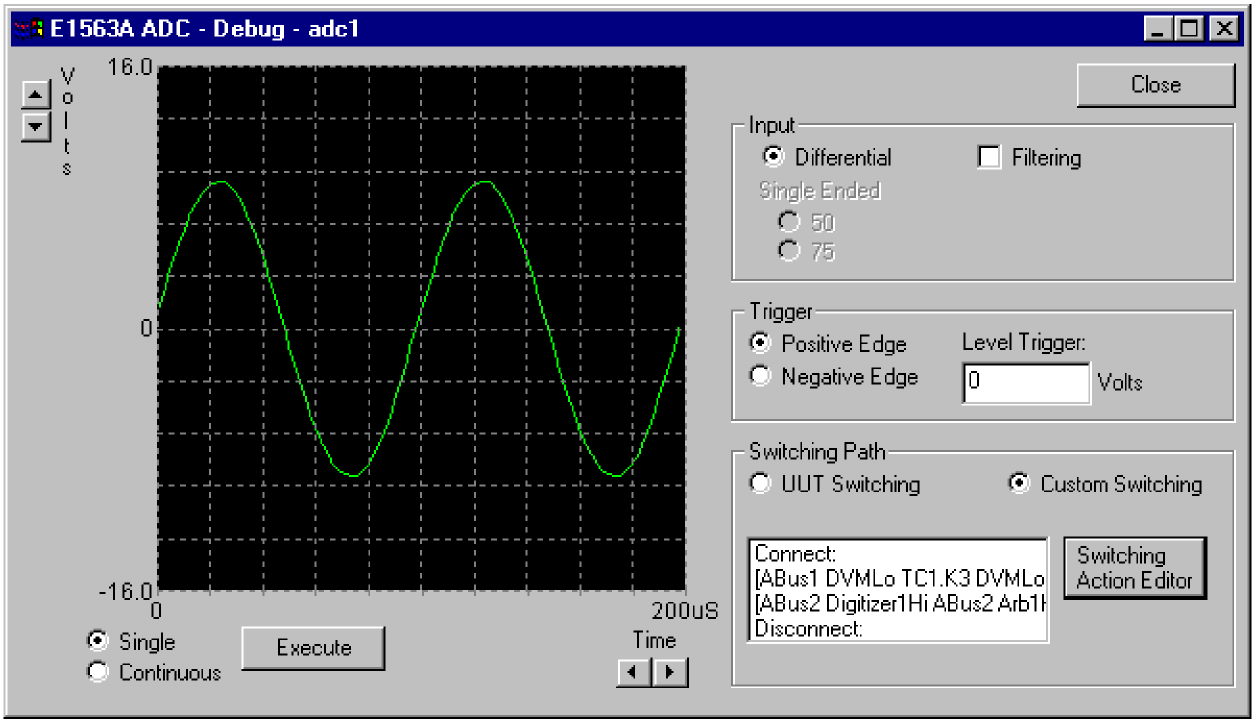


Figure 2-15. VXI Technology E1563A Digitizer Debug Front Panel

The debug panel boxes and buttons do the following:

Time Base and Range:

The range and timebase can be modified by selecting the “arrow” keys on the side and bottom of the waveform. To update the display, take a new measurement.

The time base allows the following range of selections:
    50 nS to 20 Sec in 1, 2, 5 steps
the range has the following selections:
    0.1 to 100 volts in 1, 2, 5 steps.
When the dialog box is brought up, the combo box contains the current time base and range. All of the setting for the digitizer are saved until an actual sweep is initiated.

Input and Triggering:

These two sections allow you to set a variety of input and trigger functions. The setting of each function is sent when a measurement is made.

Switching Path:

The debug panel supports the selection of a switching path. The digitizer switches both the high and low connections of the output.

Waveform Display:
The digitizer can generate either a single waveform display or a continuous update of waveforms, dependent on the selection of the "Single" and "Continuous" buttons. The display updates at a rate of >10 update/sec.

Close:
This button closes the debug panel.

System Interface

The setting of the digitizer function will use the following action routines:adcConfArm, dcConfFreq, adcConfInControls, adcIsSet, adcGetResults

Agilent E6171B Measurement Control Module

Agilent E6171B Measurement Control Module Debug Front Panel shows the dialog box for controlling the Agilent E6171B Measurement Control Module.

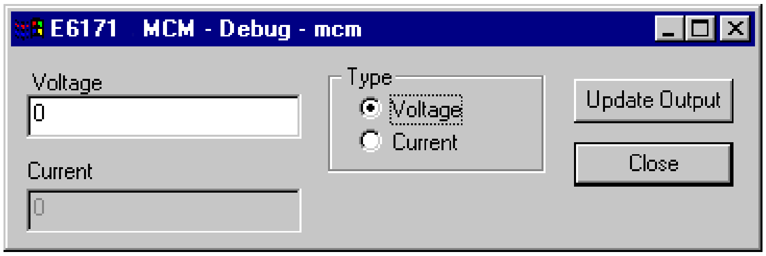


Figure 2-16. Agilent E6171B Measurement Control Module Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage or Current:
You have the option to specify the output voltage or current for each channel of the MCM. The 'Type' selection sets the voltage or current mode.

Dependent on the type selected, enter the voltage value or current value into the Voltage or Current field, respectively.

Expected Current (when enabled by the Type-Current):
Defines the expected current of the MCM (or VI, i.e., Voltage/Current) source.

Update Output:
This button changes the output source of the MCM.

Close:
This button closes the debug panel.

System Interface

The setting of the counter function will use the following action routines: viSetSourceDCI, viSetSourceDCV, viSet

Agilent E6174A Event Detector

Agilent E6174A Event Detector Debug Front Panel shows the dialog box for controlling the Event Detector Module.

Figure 2-17. Agilent E6174A Event Detector Debug Front Panel

The debug panel boxes and buttons do the following:

Clock Frequency:
Select the clock rate of the event detector module.

Edge Trigger, External Gating and Number of Events:
Specifies the type of edge triggering to be used and external gating. The text box defines the number of events to be logged before a measurement is complete.

Event Display:
The event detector can generate either a single display or a continuous update of events. The table provides a simple list view of the event and it’s time stamp. The column width can be adjust by dragging the line which separates the titles.

Execute:
This button starts a measurement.

Close:
This button closes the debug panel.

System Interface

The setting of the event detector will use the following action routines:
eventMeasure

Agilent E6198A Switch/Load Unit

Agilent E6198A Switch/Load Unit Debug Front Panel shows the dialog box for controlling the Switch/Load Unit.

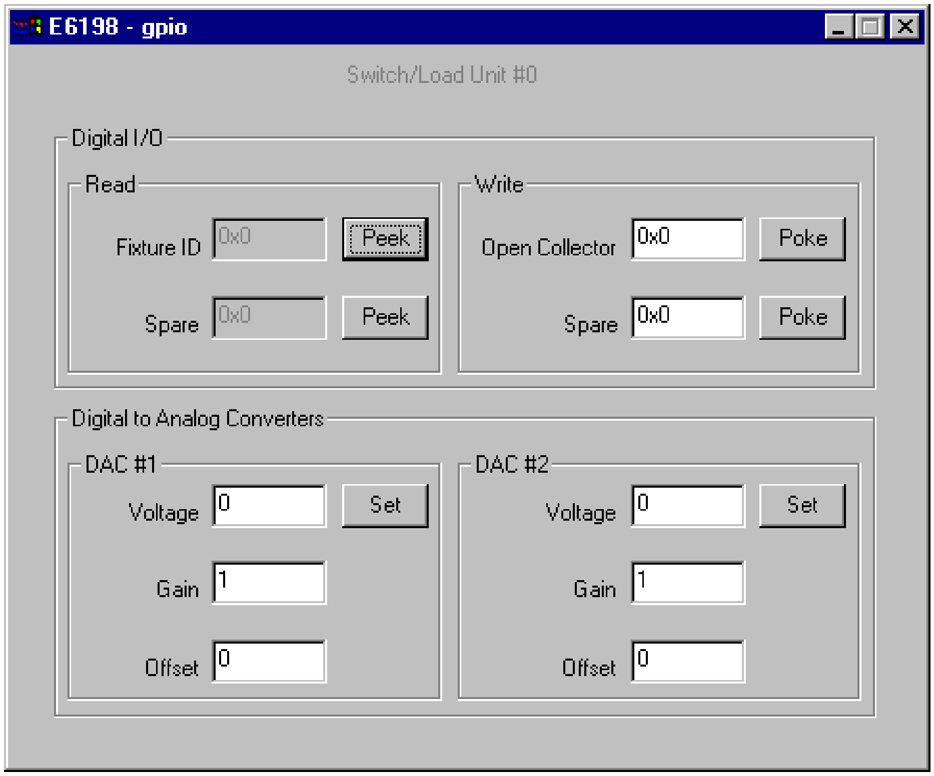


Figure 2-18. Agilent E6198A Switch/Load Unit Debug Front Panel

The debug panel boxes and buttons do the following:

Digital I/O Read:
Reads a value from the Agilent E6198 switch/load unit digital input ports. Fixture ID is value from Fix_ID(0..7) of access connector J104 and Spare is value from Spare_DigIn(0..7) of access connector J104.

Digital I/O Write:
Write a value to the Agilent E6198 switch/load unit digital output ports. Open Collector writes to output port on the switch/load unit backplane. Spare writes to output Spare_DigOut(0..7) on backplane connector J104.

Digital to Analog Converter DAC #1/DAC #2:
Change the voltage, gain, and offset values used to set the DAC voltage in the switch/load unit for both DAC #1 and DAC #2.

System Interface

The setting of the switch/load unit will use the following action routines: digitalWriteSU, digitalReadSU, dacSetDCVSU, dacSetGainOffsetSU

Agilent 6642A, 6643A, 6652A, 6653A, 6673A Power Supplies

Power Supply Debug Front Panel shows the dialog box for controlling the Agilent 6642A, 6643A, 6653A, 6673A Power Supplies. You can specify the output voltage or current for each channel of the Power Supply. You must specify a protection voltage greater than the voltage setting.

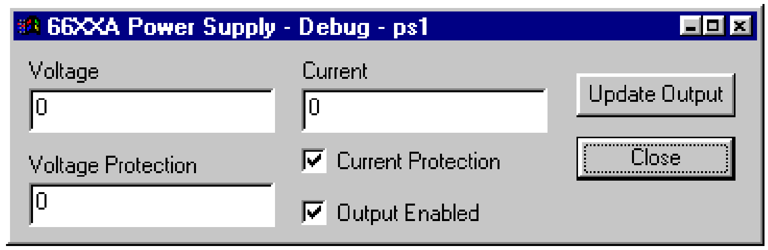


Figure 2-19. Power Supply Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage/Current:
Specifies the output voltage or current for each channel of the power supplies. The 'Type' selection sets the voltage or current mode.

Dependent on the type selected, enter the voltage value or current value into the Voltage or Current field, respectively.

Update Output:
This button changes the power supply output.

Close:
This button closes the debug panel.

System Interface

The setting of the counter function uses the following action routines:
psConfVI, psSet, psIsSet, psConnect, psDisconnect

Fast Connection Selection

You can select a switching path to a module/instrument either by selecting pins on the Unit Under Test (UUT) (see “Switching Path” box in Agilent E1411 DMM Debug Front Panel (showing Path Selection)A on page 30) or by creating a custom path using the Switching Action Editor (see “Switching Action Editor” button in Agilent E1411 DMM Debug Front Panel (showing Path Selection)B on page 30).

Specifying Unit Under Test Pins

When selecting UUT pins, you normally specify from 1 to 4 connections, depending on the module/instrument. For example, for the Digital Multimeter (DMM), you define these four connections: high, low, high current sense, and low current sense. The Counter only has only a high connection.

The software works backwards though the switching configuration and generates a switching path which connects to ABus*x*, depending on the option selected. For example, a typical connection from ABus1 to the high input of the Digital Multimeter would be: [DVMHi ABus1 VISrcHi]

Creating a User Defined Switching Path

User defined switching paths are created using the “Select Switching Path” screen which is enabled by the “Switching Action Editor” button (see Agilent E1411 DMM Debug Front Panel (showing Path Selection)B on page 30). The “Select Switching Path” screen shows two different fields, the ‘Connect’ and ‘Disconnect’ fields. All paths added or listed in the ‘Connect’ field are always connected. All paths added or listed in the ‘Disconnect’ field are always disconnected. Note that the fields are blank, if no paths have previously been added.

To add/edit/delete a path, first enable the “Select Switching Path” screen using an appropriate instrument panel, as shown in Figure 2-20. The figure uses the Agilent E1411B Digital Multimeter instrument panel.

The following shows how to add/edit/delete a path.

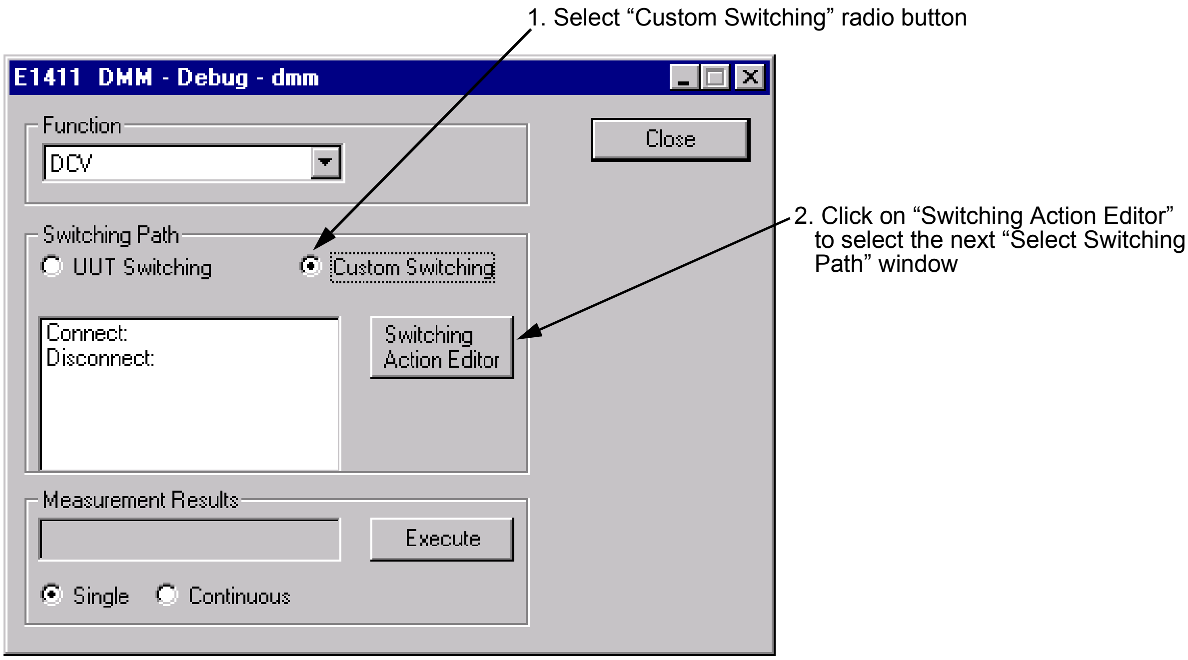


Figure 2-20. Enabling the “Select Switching Path” Screen

Adding a New Path

To add a new path, first determine the end nodes or points in the path. Then use the “Switching Action Editor” to find a path using the intermediate nodes between the end notes or points.

For example, to connect the Hi input of the Agilent E1411B Digital Multimeter (DMM) to the Hi output of the Agilent E6171 Measurement Control Module (MCM), you must know the node names for these connections. Also, since the DMM has no direct connections to the VI source or MCM, it must connect using ABus*x*. Thus, the nodes for this path would be: [DVMHi ABus1 VISrcHi].

Before adding a new path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen. Then use the procedure in Figure 2-21 to add the path.

Editing a Switching Path

To edit a path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen, if the screen is not enabled. Then use the procedure in Figure 2-22 to edit the path.

Deleting a Path

To delete a path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen, if the screen is not enabled. Then use the procedure in Figure 2-23 to delete the path.

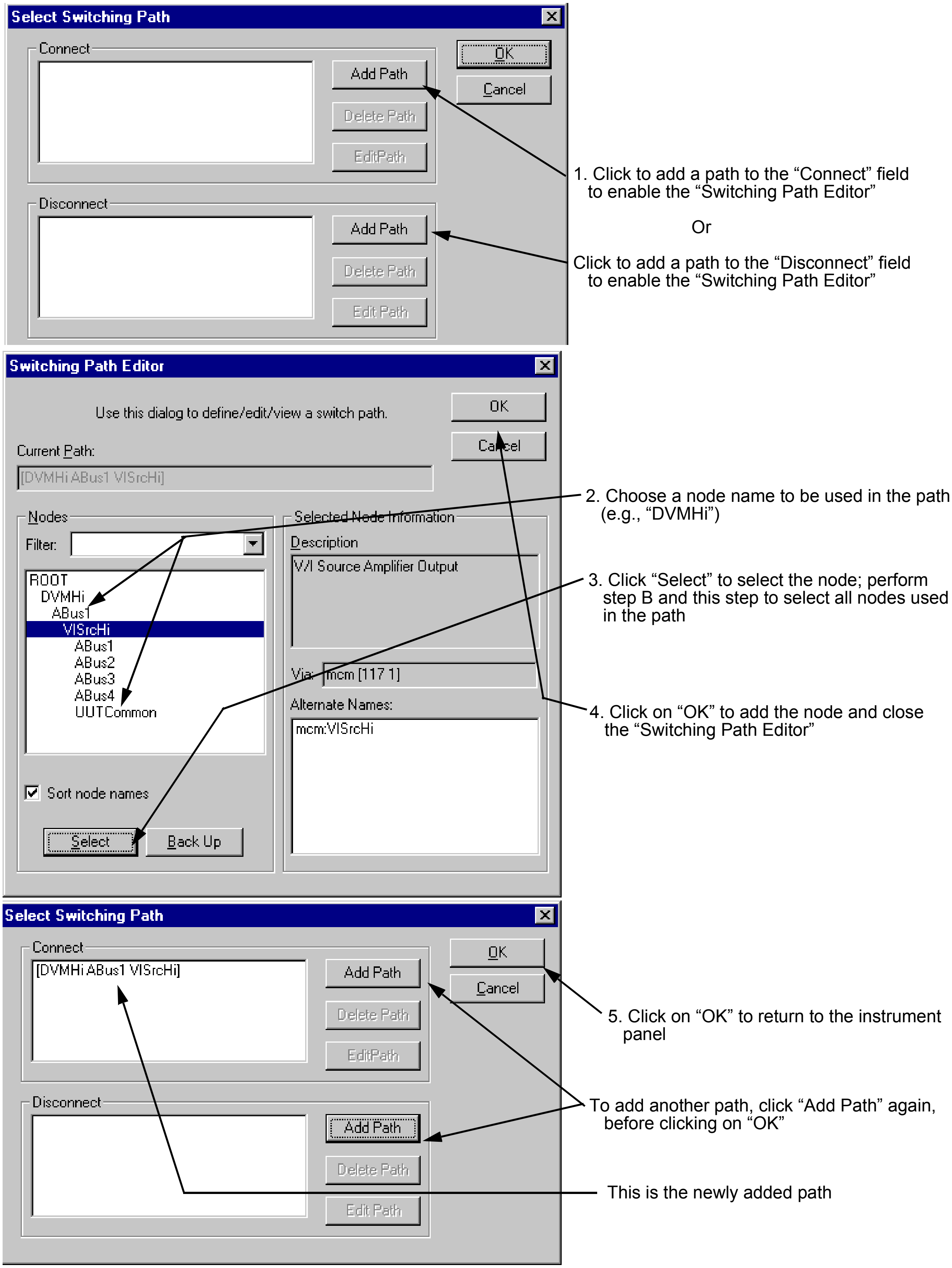


Figure 2-21. Adding a New Switching Path

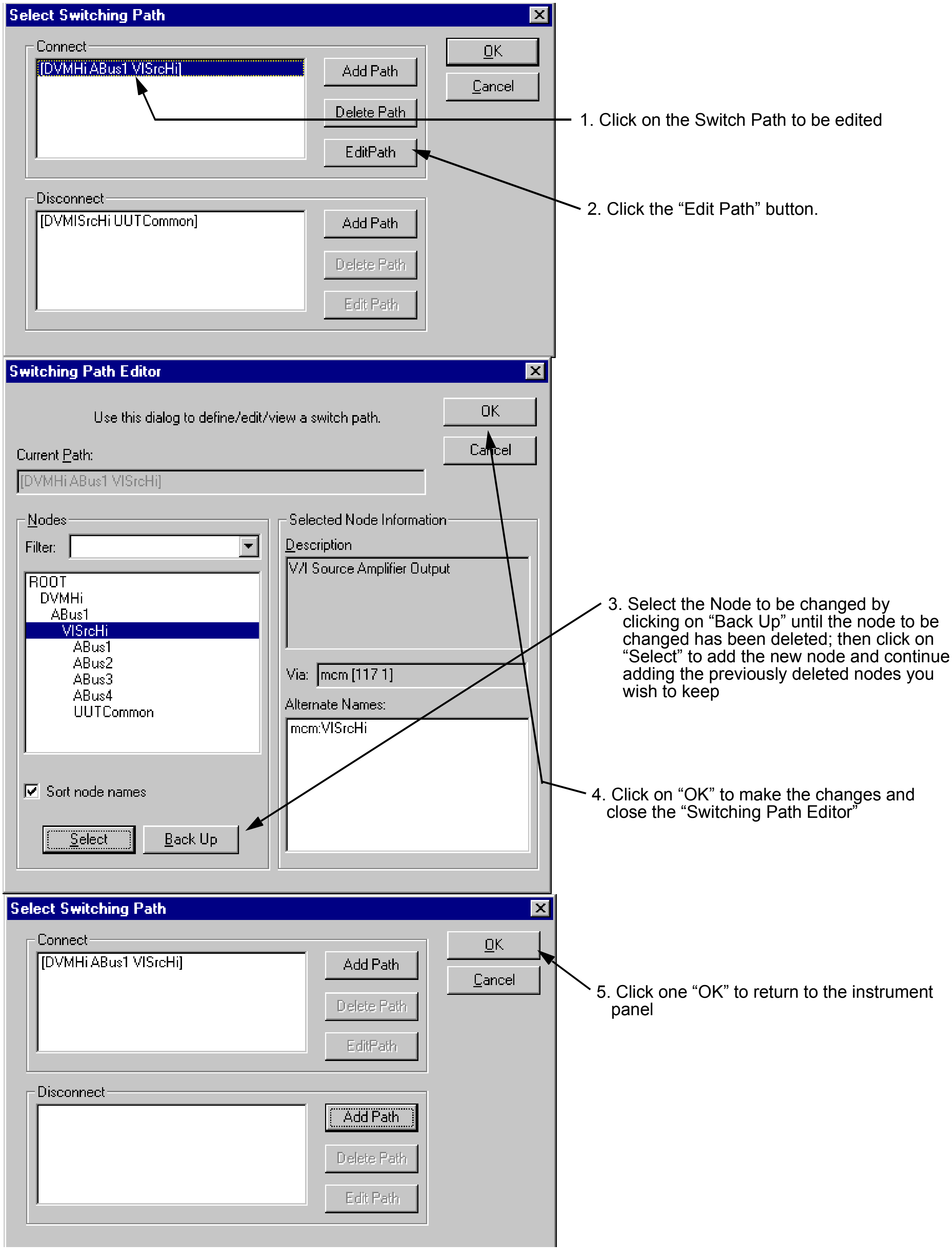


Figure 2-22. Editing a Switching Path

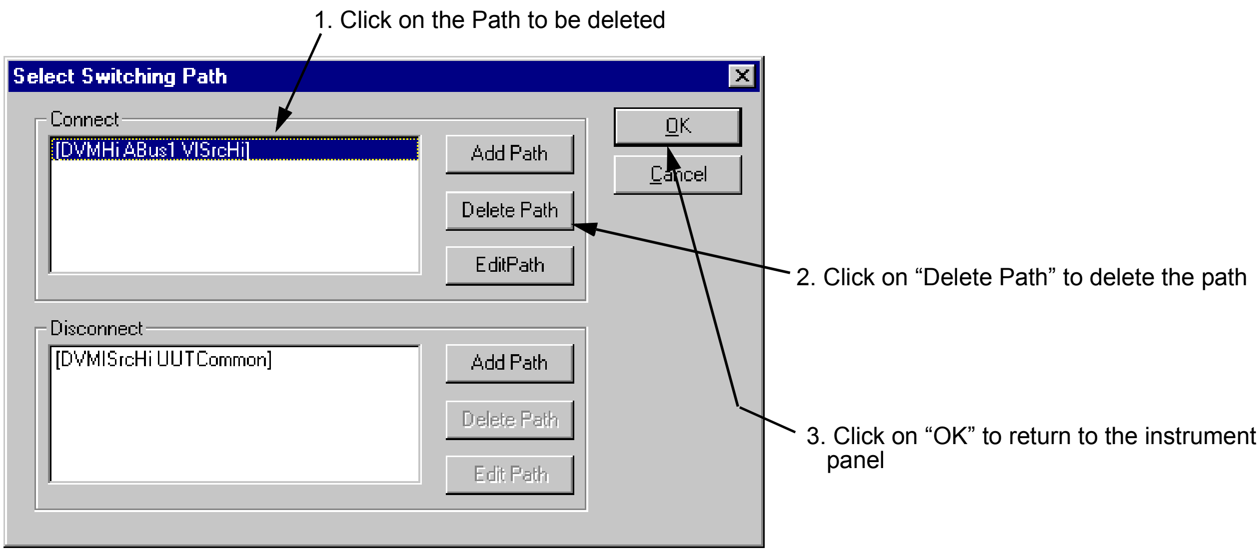


Figure 2-23. Deleting a Switching Path

Viewing Waveforms

A ‘Waveform’ data type is available in the “ADC_Analyze_Wave” High-Level Action. This data type includes a graphical viewer for data. Sample Testplan Showing Waveform Data Type shows a sample testplan using the “ADC_Analyze_Wave” Action with the Waveform parameter. Range Data on the Waveform Display Graphical Editor and Sample Waveform Display show the waveform editor, and Figure 2-27 shows the waveform display.

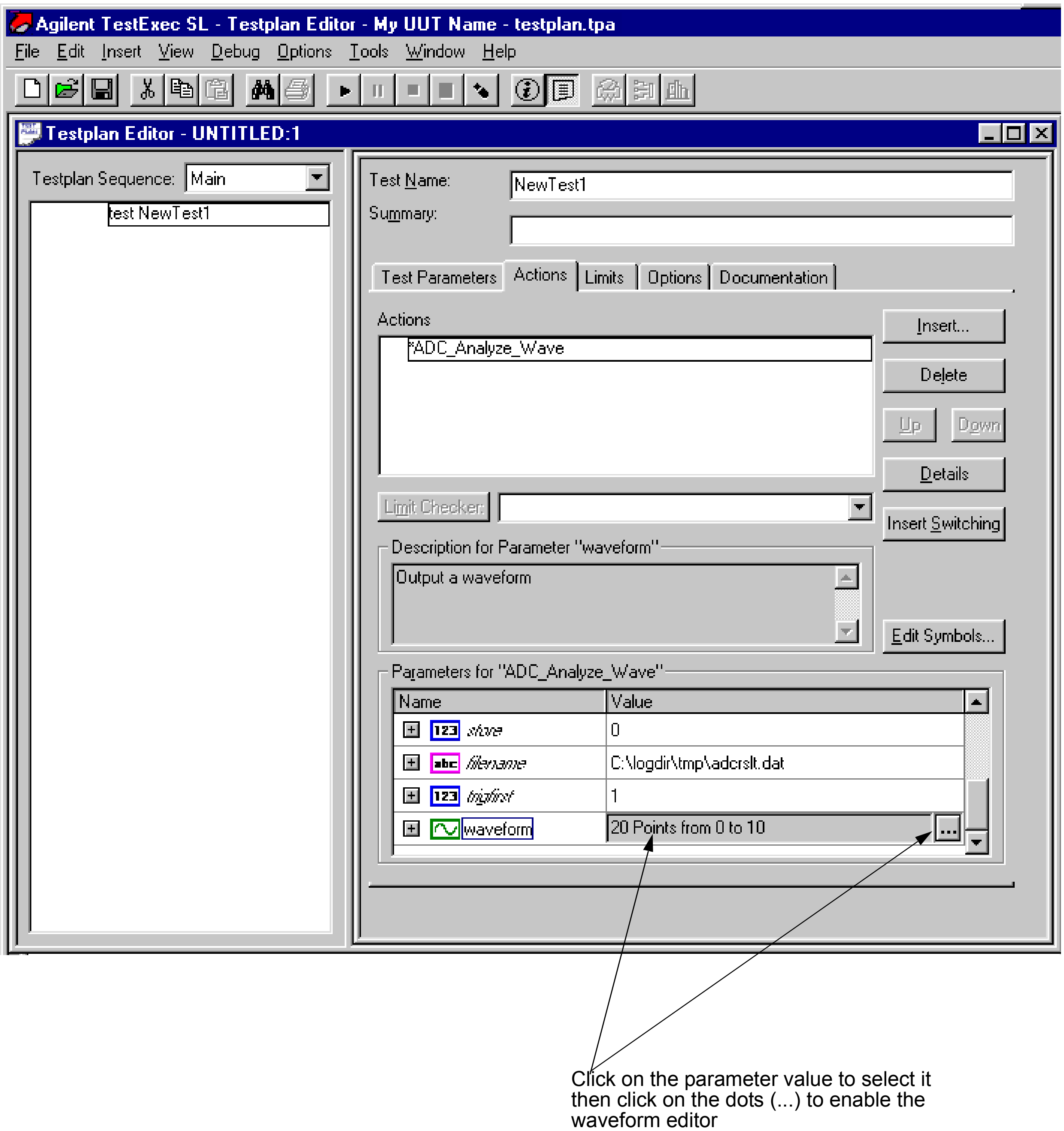


Figure 2-24. Sample Testplan Showing Waveform Data Type

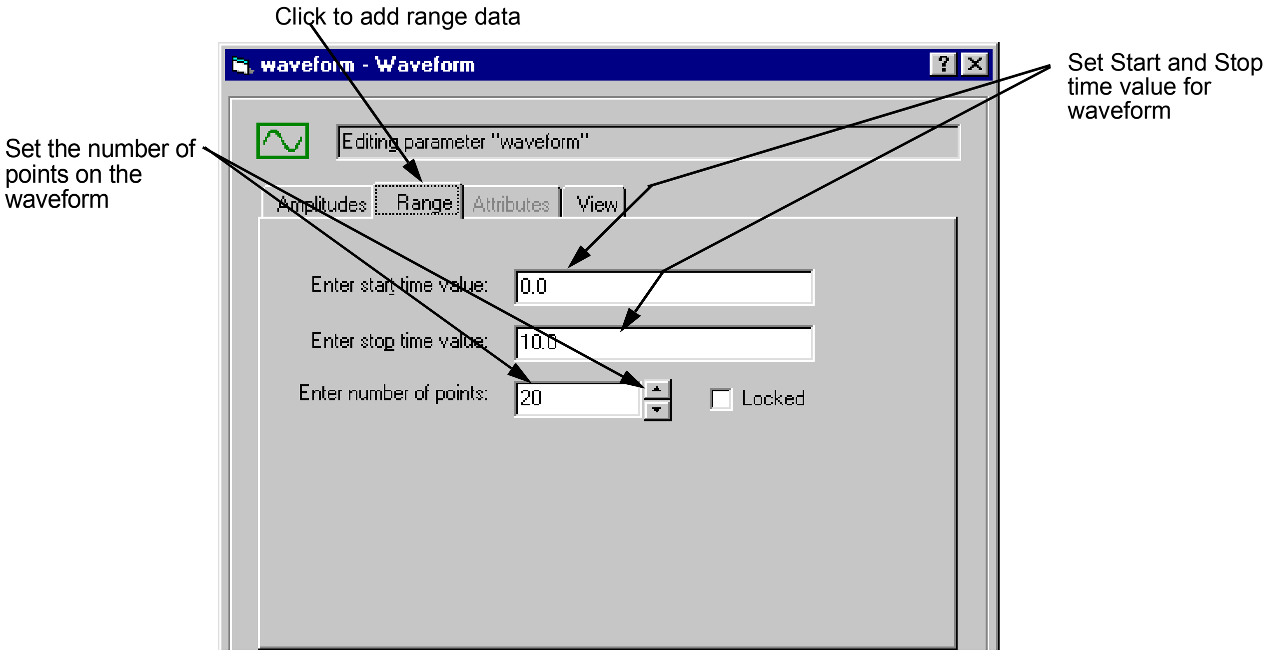


Figure 2-25. Range Data on the Waveform Display Graphical Editor

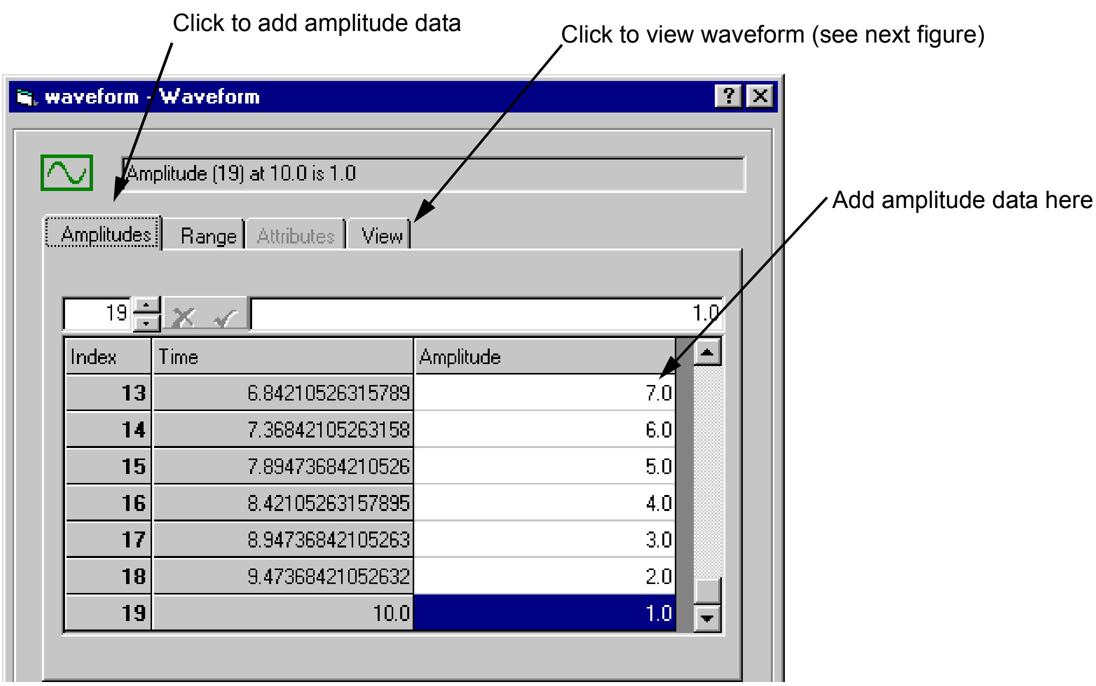


Figure 2-26. Amplitude Data on the Waveform Display Graphical Editor

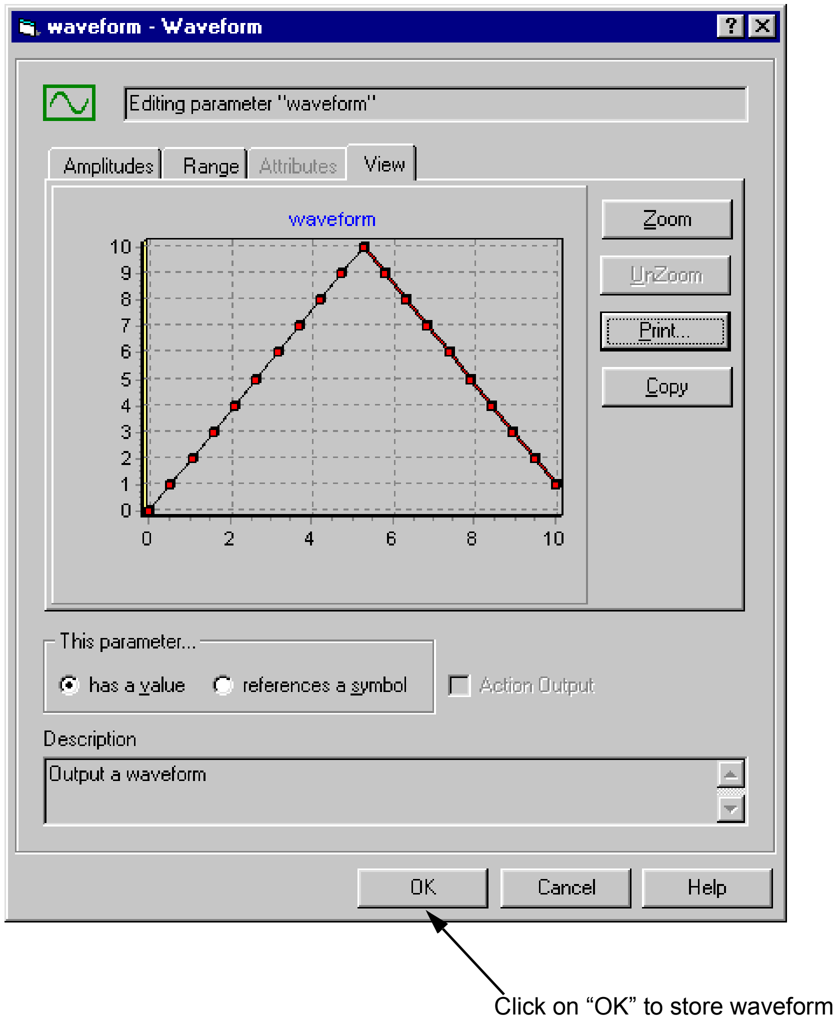


Figure 2-27. Sample Waveform Display

Instrument Handlers

Instrument Handlers are a layer of software between Agilent TestExec SL and standard instrument drivers (see Figure 2-28). In general, Instrument Handlers are designed to be called from C/C++ code action

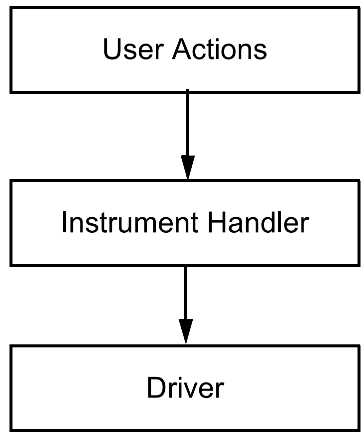


Figure 2-28. Software Layers

Instrument handlers contain functions written in C code that are organized by instrument type and function, and require parameters relevant to the function. For example, the call to set up triggering for a voltmeter is:

```
dmmConfTrigIn (dmm, trigselect, count, delay);
```

Function `dmmConfTrigIn` accepts four parameters: dmm, trigselect, count, and delay. The generic name of the function is `ConfTrigIn`, and the name of the instrument, in this case `dmm`, is added as a prefix to form the full, specific name of the function.

Note Instrument handlers are maintained constant in different TS-5400 software releases, which is not the case with the drivers. Thus, use instrument handlers for actions whenever possible.

Using the Action Wizard to Develop Actions

The Agilent TestExec SL application used in the TS-5400 System comes with a program called Action Wizard to develop actions. The Action Wizard automatically runs through the steps necessary to develop an action. The wizard also includes a help file with more detailed instructions and information on how to use the wizard.

To run the “Action Wizard”, use the procedure in Figure 2-29.

Note Actions are the smallest components of a test and are used to setup and execute instruments, perform cleanup functions, and to make measurements.

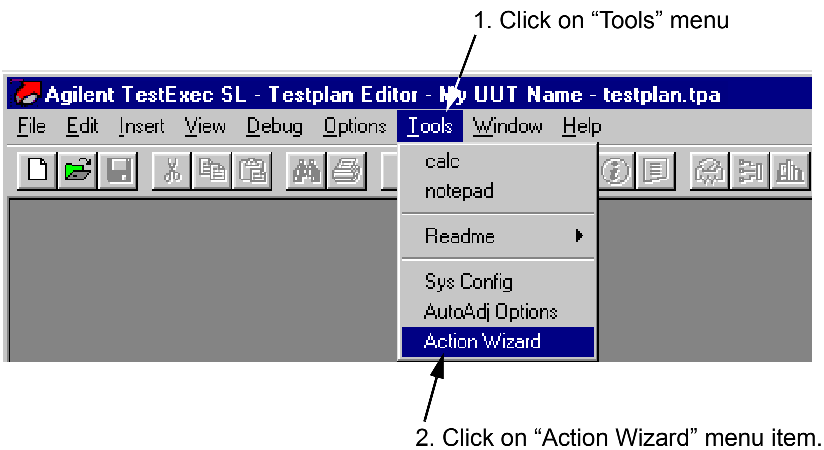


Figure 2-29. Selecting and Running the Action Wizard

Chapter 3
Planning System Testplans

Testplans are used by Agilent TestExec SL and the other system software to run tests using the Agilent TS-5400 System. Use the information in this chapter to help you plan testplans for your tests. This chapter is separated as follows:

The steps of the testplan planning and fixturing process are:

It is assumed that your system has been configured and setup at the factory with all the appropriate test instruments.

The process and tools for actually developing and fine tuning the test are documented in the *Agilent TestExec SL User's Guide* and the on-line help. These tools include test libraries, action libraries, the Switching Configuration Editor, the Switch Path Editor, and the Action Definition Editor.

Planning The Tests

For each test or each type of test required by the test specifications, complete a Test Description Worksheet. Figure 3-1 shows an example Test Description Worksheet for a simple input leakage test on a module input pin.

In many cases there is more than one way to perform a test. For example, to test injector outputs you can use a counter to measure timing and the Digital Multimeter to measure voltage, or you can use a digitizer and analyze the results for both measurements. Which test strategy you choose depends on your measurement resources and test execution time trade-offs.

Agilent TS-5400 Test Description Worksheet
Test Name Page____of____
Test Description: *Measure Input pin leakage current*
Measurement(s): (measure, limits) *Leakage low = 0.8mA, hifh = 1.2mA*
Preconditions:
UUT Setup *N/A*
Power (Volts, Amps, Pin) *Vbatt 14V, 0.2A*
Load (Value, Power, Pin) *N/A*
Constraints:
Timing
Test Sequencing
Reuse:
Test Templates
Actions: ForceVmeasI
Instrument & Measurements: (name, settings) *V/I 10V Dmm expect 2mA*
Connections: (from - to)
Cleanup:

Figure 3-1. Sample Agilent TS-5400 Test Description Worksheet

Agilent TS-5400 Test Description Worksheet

Test Description: In the test description section, briefly describe the test to be performed.

Measurements: Under measurements, describe what pin on the Unit Under Test (UUT) will be measured, the expected measurement, and what the test limits are. The Agilent TestExec SL software allows multiple results to be limits checked.

Preconditions: For the Unit Under Test, use the preconditions section to describe the state the UUT must be in before performing the measurement as well as how to set it to the desired state. For power, describe what power connections are required for this test. Under the loads portion of the preconditions section, describe what loads, if any, are required to perform the test.

Constraints: Under constraints give information on timing constraints such as settling time before a measurement is valid, as well as test order constraints such as making sure the serial interface has been tested before using it to set up the UUT in this test.

Reuse: To complete the reuse area, use the test executive's Action Libraries box and Test Libraries box to find actions and tests, respectively, that can be reused.

Instruments & Measurements: The instrument section is used to describe what instruments are needed for the test and how they should be programmed.

Connections: In the connections section, describe how the resource will be connected (e.g., through a 32-pin Matrix module, directly, or indirectly through the Matrix module auxiliary channels connector). The 32-Pin Matrix module has 16 external instrument connections which can be connected to four analog buses (AB1 to AB4) or the UUT common ground bus. The Digital Multimeter, one channel of the Arbitrary Waveform Generator, one or more channels of the DAC, the Counter, the Digitizer, and the Measurement Control module resources are generally connected to the UUT through Matrix modules. Loads in the load box are typically connected to the UUT through the TC1 - TC8 connectors (Test System Interface) or the one-one-two expander blocks (Mass Interconnect).

The 32-Pin Matrix modules can handle signals up to 200 volts peak. Other instruments may have even lower voltage restrictions.

Caution **Input signals greater than 100 $V_{peak/dc}$ voltages can damage the digitizer. Input signals greater than 200 $V_{peak/dc}$ voltages can damage the 32-Pin Matrix modules. Other instruments may have lower voltage restrictions. Voltages exceeding these limits must be attenuated before being applied to these modules to avoid damage.**

Cleanup: Cleanup describes the state the system and UUT should go to upon completion of the test

Planning The Resources And Fixture

The goal of the resource planning step is to determine the use and connect strategy for instruments and loads. The result is a description of the Test System Interface/Mass Interconnect fixture that makes the connections between the test system and the UUT.

Determining System Resources

Before planning the fixture you must know what resources are available in the system and where you can access them. The Agilent TS-5400 System Description Worksheet organizes this information. The worksheet should be completed before the first fixture is built and then added to when hardware is added to the system. When adding hardware, it is important to make sure the new hardware will not interfere with an already existing test or fixture.

Agilent TS-5400 System Description Worksheet

Figure 3-2 shows an example Agilent TS-5400 System Description Worksheet using the Test System Interface and Figure 3-3 shows a worksheet for the Mass Interconnect.

Agilent TS-5400 Test Description Worksheet using a Test System Interface

ICA Description Table

Connector #	Notes
TC1	Pin Matrix Module #1
TC2	Pin Matrix Module #2
TC3	Pin Matrix Module #3
TC1	Load Cards
TC2	Load Cards

Load Wiring Table

Slot #	Card Type	Channel #	Value	Current Sense Resistor Value	ICA ID	No Load Bus	NC Load Bus	8 CH Bridge Jumper
1	8 Channel	1	short	50 milliohms	11A1	PB2		
		2	short	50 milliohms	11A3		PB1	
		3	1 k ohm	50 milliohms	11A5	PB2	PB1	
		4	1 k ohm	50 milliohms	11A7	PB2	PB1	

Figure 3-2. Example Agilent TS-5400 System Description Worksheet using a Test System Interface

Agilent TS-5400 Test Description Worksheet using a Mass Interconnect

ICA Description Table

Slot #	Connector	Notes
1	91421	Pin Matrix Module #1
2	91421	Pin Matrix Module #2
3	91421	Pin Matrix Module #3
11	91423	Load Cards
12	91423	Load Cards

Load Wiring Table

Slot #	Card Type	Channel #	Value	Current Sense Resistor Value	ICA ID	No Load Bus	NC Load Bus	8 CH Bridge Jumper
1	8 Channel	1	short	50 milliohms	11A1	PB2		
		2	short	50 milliohms	11A3		PB1	
		3	1 k ohm	50 milliohms	11A5	PB2	PB1	
		4	1 k ohm	50 milliohms	11A7	PB2	PB1	

Figure 3-3. Example Agilent TS-5400 System Description Worksheet using a Mass interconnect

ICA Description Table

The ICA Description Table in Figure 3-2 shows the connectors used for the Test System Interface and Figure 3-3 shows the connector type and slots used for the Interface Connector Assembly (ICA) of the Mass Interconnect. This helps determine what connector is needed for the fixture to mate with the connectors on the Test System Interface or Mass Interconnect. This information is mainly for the fixture builder, but can be helpful in determining how to connect resources.

Connector # (Test System Interface only) - the connector number used for the connections to the Test System Interface.

Slot # (Mass Interconnect only) - the ICA slot number of the Mass Interconnect used for the connector.

Connector (Mass Interconnect only) - the product number of the connector used by the Mass Interconnect (i.e.,Agilent 91421B for the general-purpose connector and Agilent 91423A for the high-power (coax) connector). The high-power (coax) connector is used for high-power connections, typically loads. The general-purpose connector is used for all other applications.

Notes - The notes section is used to describe what resources are available on the connector.

Load Wiring Table

This table describes the loads in the load box. Check this table to determine if the load required for a test is already configured in the load box. If not, add the description of the new load that will need to be added to the system. For instructions on connecting the loads, refer to the *Agilent Switch/Load Unit User's Manual*. Each column of the table is described below.

Slot # - describes the load cards position in the Switch/Load Unit.

Card Type - describes the type of load card: 8-channel, 16-channel, or 24-channel. For a description of the three different cards and their capabilities, see the *Agilent Switch/Load Unit User's Manual.*

Channel # - the number of the channel on the load card.

Value - enter the load value (resistance, capacitance, inductance) or the load device.

Current Sense Resistor Value - the value of the current sense resistor configured in the load. The factory installed default value is 0.050Ω.

ICA ID - the ICA connection for this load.

NO Load Bus - describes where the normally open connection has been jumpered. This is frequently a power bus.

NC Load Bus - describes where the normally closed connection has been jumpered. This is frequently the ground power bus.

8 Ch Bridge Jumper - shows which channels have been jumpered together if the Agilent E6175A 8-Channel load card is used in a bridge configuration.

Assigning Resources

To assign the resources for a testplan, you need to complete the Agilent TS-5400 Fixturing Worksheet (described later in this chapter) by looking at each test individually and performing the following steps.

Note

The Test System Interface uses the nomenclature: `TCn.RowCol` where *n* is either 1 or 2 (TC1 or TC2), `Row` is A through CC, and `Col` is column 1 through 6. For example, TC2.Z3 is Test Connector 2, row Z, column 3. Refer to the *Express Connect Test System Interface Wiring Guide* for more information.

The Mass Interconnect uses the following nomenclature to identify a specific pin the mass interconnect: `ColConPin` where `Col` is the column of the mass interconnect ITA (1 to 21), `Con` is either the top connector (A) or the bottom connector (B), and `Pin` is the actual pin number (1 - 96). Since the high-power (coax) connectors only have one grouping of connections, always use connector A to describe the connection. For example, 2A7 refers to the second column, top connector, and 7th pin. Refer to the *Agilent E6170 Mass Interconnect For the Agilent TS-5400 Test System User's Manual* for more information.

---

1. Enter required resources, for each UUT pin used in the test, in the Load Value and Other Resource columns of the worksheet.

2. Look at the resource connection information listed in the Test Description Worksheet and describe it in the corresponding mass interconnect ID columns in the fixture table.

Look at the System Topology Layer of the Agilent TestExec SL Switching Configuration Editor for ICA IDs in your system.

Thought needs to be given to Matrix module allocation. If Matrix module resources are not in short supply, it may be helpful to wire a Matrix module resource to each UUT pin. This allows the most flexibility to handle board design or testing strategy changes. If you do not have enough Matrix module channels to assign one to each UUT pin, you will need to be more selective in making the assignments.

For a UUT ground pin, a Matrix module UUT Common pin, signal pin, and earth ground should be wired to connect the system's UUT Common and UUT ground, to earth ground and provide a way to allow sensing.

3. If the test requires a load which is not currently configured in the load box, analyze the load requirements for peak current and load size to determine which type of load card to use. Then add the description of the load to the Load Wiring Table of the Agilent TS-5400 System Description Worksheet.

Agilent TS-5400 Fixturing Worksheet

While completing the Agilent TS-5400 Fixturing Worksheet, remember that this worksheet will be used to build the fixture. It is important to be consistent in your naming conventions and the way that information is entered in the form. Suggestions are given in the descriptions of the worksheet columns. You can also refer to “Wiring the Fixture” later in this chapter.

Figure 3-4 shows the Agilent Fixturing Worksheet. Each section of the worksheet is described below.

Note The worksheet is for a system using the Mass Interconnect. Systems using the Test System Interface is similar, but with different connector numbers and with no “Expander Blocks”.

Agilent TS-5400 Test Description Worksheet

Fixture Table -- UUT Connections

UUT Pin #	UUT Pin Name	Matrix Card ICA ID	Load Value	Load ICA ID	Other Resources	Other Res. ICA ID	Expander Block	Notes
1	Gnd	2A3	Short	1A3				
		2A1						Shield Gnd
		2A4						Shield Gnd
		2A4						Shield Gnd
		2A10						Shield Gnd
		2A13						Shield Gnd
		2A16						Shield Gnd
		2A19						
2	VBatt	2A6	Short	11A1				
3	Ignition	2A9						
4	Output	2A12	1 k ohm	11A5				
5	Serial In	2A15				Com2 Out	1A2	
6	Serial Out	2A18				Com2 In	1A2	
7	Injector	2A21	1 k ohm	11A7				

Fixture Table -- Jumpers

	Jumper Name	From ICA ID	To ICA ID
		2A93	1B43
		2A96	1B45

Figure 3-4. Example Agilent TS-5400 Fixturing Worksheet

Fixture Table - UUT Connections

This section describes the connections that will be made to the UUT.

UUT Pin # - the physical name of each connection to the UUT. They should be listed in order making sure not to skip any pins, including pins which are not connected. The typical format for the UUT pin number is J1-1. This corresponds to Jack 1-pin 1. You should leave several lines between each entry to allow for multiple resource allocation entries.

UUT Pin Name - shows the logical name corresponding to each UUT pin number. The Agilent TestExec SL software will allow you to enter this node name information and build testplans using the symbolic node name (alias). (See on-line help for the Switching Configuration Editor)

Matrix Module ICA ID - lists the ICA IDs of assigned Matrix module resources. The fixture builder will attach a wire (either directly or through an expander block) between the UUT Pin # and the ICA ID location called out in this column.

Load Value - enter the load value (resistance, capacitance, inductance, wattage) or the load device which must be connected to the UUT pin for each test.

Load ICA ID - determine the ICA connection point for the appropriate load and enter it in this column. The fixture builder will attach a wire (either directly or through an expander block) between the UUT Pin # and the ICA ID location called out in this column.

Other Resource - used to describe required resources, other than loads, which cannot be accessed through Matrix modules.

Other Res. ICA ID - describes the connection to the UUT from resources other than loads and Matrix modules. The fixture builder will attach a wire (either directly or through an expander block) between the UUT Pin # and the ICA ID location called out in this column.

Expander Block - specifies that connections to or from a certain UUT pin be made through an expander block. Two types of expander blocks are available. The blocks allow you to connect many ICA points to a UUT pin or to connect many UUT pins to a single ICA location without having many connections at one point. As a general rule an expander block should be used whenever more than three connections need to be made to the same point. For reliable connections, no more than two wires should be connected to each terminal of an expander block.

Place a check mark in this column if all connections to this UUT point should be made through an expander block. If an expander block will be used to connect one resource to several UUT pin numbers, write the other UUT pin numbers in this column.

Notes - allows you to clarify any of the other fields.

Fixture Table - Jumpers

This section is used to describe jumpers in the fixture. It is also the place to describe any fixture electronics including the safety interlock (not used in the Test System Interface) and fixture identification jumpers. The safety interlock is a normally-open switch which shorts pins 1A13 and 1A15 when the fixture and safety shroud, if needed, are present. The fixture identification jumpers are placed in rows 2 through 5 of connector 1B to give the fixture a unique binary number for identification purposes. This feature is not implemented in software release A.01.00. For each jumper, give the jumper name and the From and To ICA ID.

Notes:

Chapter 4

Using the DMM and ARB

Chapter Contents

This chapter describes how to trigger the DMM and how to use the ARB. The chapter is separated as follows:

Triggered Voltage Measurement

The Agilent E1411B Multimeter (DMM) can be triggered by a signal on the VXI Mainframe's backplane. This signal uses one of the TTL (i.e., TTL0 to TTL7) trigger lines.

How to Trigger the DMM

Testplan '`dmmtrig.tpa`' shows how the DMM is triggered using the Agilent E6174A Arbitrary Waveform Generator (Arb). The Arb is also used to generate the signal measured by the DMM. The testplan also demonstrates how to use the Agilent E6171B Measurement Control Module's (MCM) analog comparator. The testplan is in the following directory:

`C:\Program Files\Agilent\TS-5400 System Software\testplan\examples`

The Arb generates a ±5V 10 Hz square wave as the trigger signal. This method uses the analog comparator of the MCM to send the trigger to the DMM using the VXI Mainframe's backplane (see connections in Figure 4-1 and Figure 4-2).

The testplan configures the MCM analog comparator to trigger at 0 volts. Setting the "trigedge" parameter in the "viConfCompare" action to "0" (positive edge detection), causes negative going backplane trigger when the input signal makes a low to high (positive going) transition through the trigger threshold. Setting the "trigedge" parameter in the "viConfCompare" action set to "1" (negative edge detection), causes triggers when the input signal makes a high to low (negative going) transition.

The following is a review of the testplan

Test/Action Name	Description
test: Dmm Init	Initializes the Digital Multimeter.
dmmConfCal	Disables autozero and set the power line frequency.
dmmIsSet	Waits for setup to complete.
test: Arb 1 pgm	Configures Arb and outputs a 10 Hz +5V Square wave.
arbReset	Reset the Arbitrary Waveform Generator (Arb) to its turn-on state.
arbConfOutControls	Configures the Arb's output circuitry.
Arb_Dl_Std_Waveform	Download a 10 Hz square wave into the Arb.
arbInitiate	Start outputting the waveform.
Arb_Select_Wave_By_Name	Select waveform downloaded above.
test: VI Config0 (+5)	Configures MCM's analog comparator to trigger at 0V and a low to high (positive transition) trigger.
viConfCompare	Configures the comparator. The "trigedge" parameter value (0) causes a backplane trigger when the input signal makes a low to high (positive going) transition trough trigger threshold.
viIsSet	Waits for setup to complete.
Test: Meas V - Trigger	Trigger DMM and measure trigger voltage.

Test/Action Name	Description
Switching	Make the necessary switching to setup the Arb to DMM measurements and trigger signal paths.
dmmMeasureTrigVoltage	Configures DMM for a measurement using the backplane TTL trigger line 0.
dmmGetResults	Make the measurement and return the reading.
test: VI Config1 (-5)	Configures MCM's analog comparator to trigger at 0V and a high to low (negative transition) trigger.
viConfCompare	Configures the comparator. The "trigedge" parameter value (0) causes a negative going backplane trigger when the input signal makes a high to low (negative going) transition trough trigger threshold.
viIsSet	Waits for setup to complete.
test Meas V - Expanded	Trigger DMM and measure trigger voltage.
Switching	Make the necessary switching to setup the Arb to DMM measurements and trigger signal paths.
DelayMillisecond	Wait for switching to complete.
dmmInitiate	Initiates a reading cycle.
dmmGetResults	Make the measurement and return the reading.

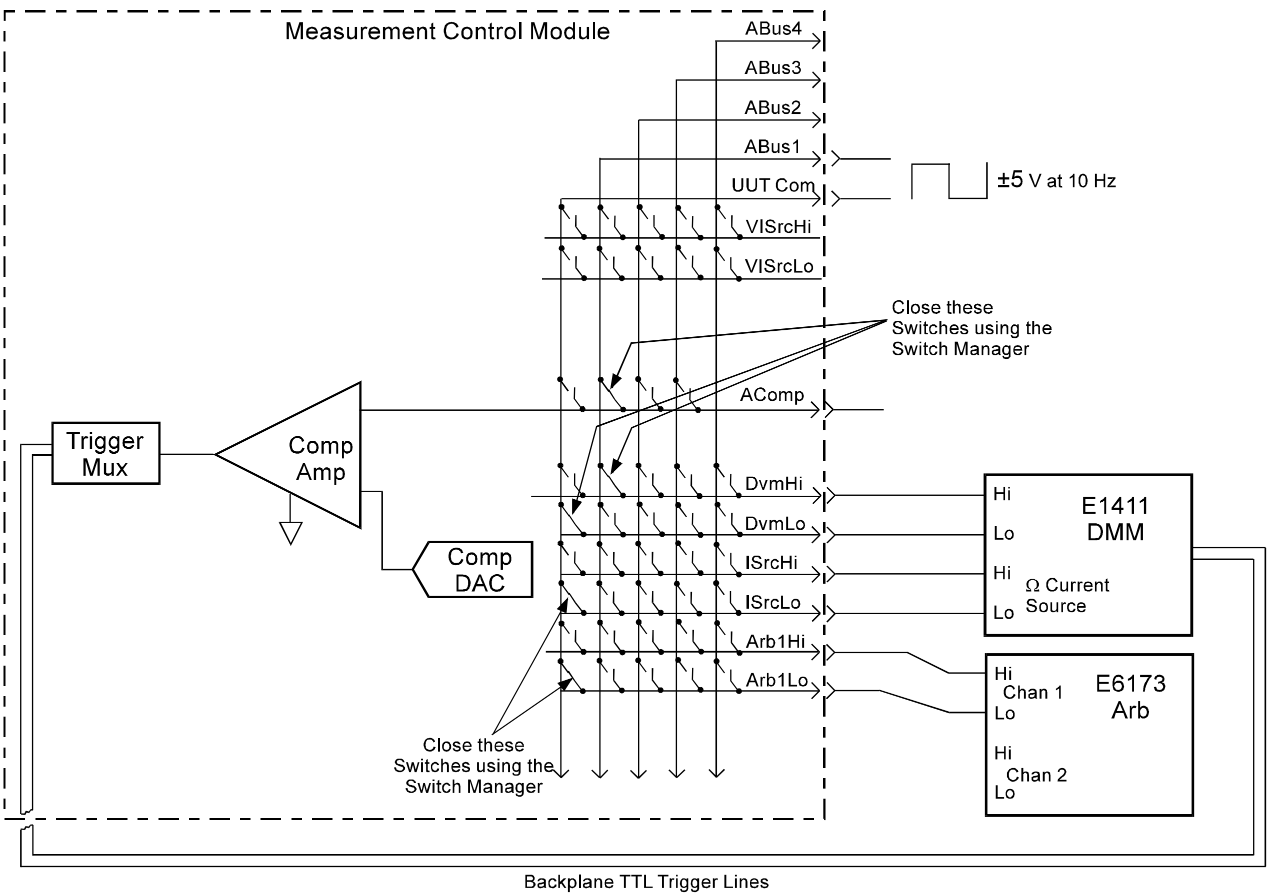


Figure 4-1. Trigger Connections

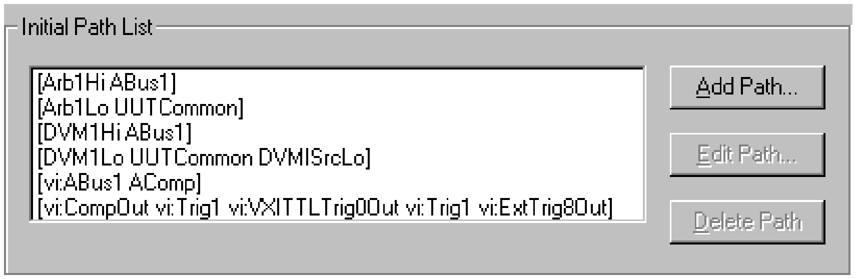


Figure 4-2. Switch Paths Listing

Using the Agilent E6173A Arbitrary Waveform Generator

The following describes the Agilent E6173A Arbitrary Waveform generator and how to use it.

Arbitrary Waveform Generator Description

The Agilent E6173A Arbitrary Waveform Generator (Arb), formerly the Agilent Z2471A, is a register-based, two channel signal generator whose channels are isolated from each other and from ground. The factory default configuration connects first channel to the Agilent E6171 Measurement Control Module (MCM).

You can amplify channel one's output using the MCM's V/I amplifier/current source. You can connect channel two directly to your Unit Under Test (UUT) or use one of the unassigned MCM inputs. The Arb can also switch to any UUT pin by switching to Aux ports, if using the default system pins.

The Arb can output standard waveforms, such as sine, square, triangle, knock, or crank of a specified frequency, amplitude, and offset, and custom arbitrary waveforms defined by the user.

How the Arb Generates Waveforms

The Arb generates waveforms using pre-defined or user defined segment and sequence data. Figure 4-3 shows a typical example of a waveform.

The segments of a waveform are the voltage levels of each point on the waveform. The sequence consists of a group of pre-defined or user defined number of segments. To generate a waveform, the Arb outputs the sequence, consisting of the defined number of segments, using a pre-defined time.

The number of segments output (i.e., how often they occur) and the time of the Arb’s internal clock (i.e., segment duration time) determines the time or frequency of the waveform to be output. The Arb’s internal clock time is 2 µS. Either single or multiple clock cycles can be used to set the appropriate frequency. Selecting the clock cycles is usually known as selecting the “dwell” count.

For example, a waveform consisting of 250 segments using two 2 uS clock cycles (or 2 dwell counts), outputs a 1 kHz waveform, shown as follows:

$$250 * (0.000002 * 2) = 250 * 0.000004 = 0.001 \text{ S} = 1 \text{ kHz}$$

In addition to the 2 µS clock, the Arb also has a 100 nS counter that can be used to add time in 100 nS steps to the 2 µS clock cycles. This is used to generate frequency accurate waveforms.The counter is normally set using the “clock adjust” function. Note: Segment time = (Dwell + 1) * (2 µS + (clock adjust * 0.1 µS))

Both the segment data and sequence data is *downloaded* into the Arb’s memory using the standard actions supplied with the system. These actions can be used to download both standard (pre-defined) waveforms and user defined waveforms.

Note For more information about the waveform timing, refer to the *Agilent E6173A Arbitrary Waveform Generator User's Manual*.

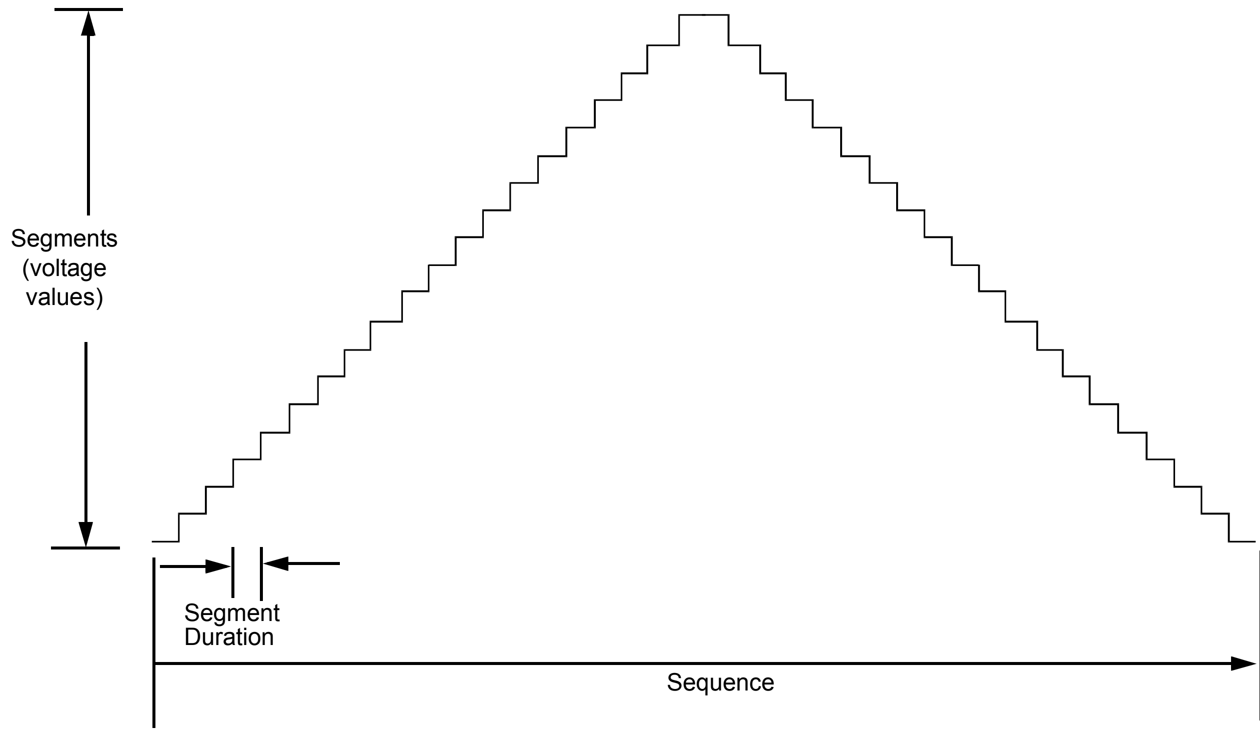


Figure 4-3. Typical Waveform using Segments and Sequence

Generating Arb Waveforms

There are several methods to generate Arb waveforms using the standard actions. You can generate standard waveforms, such as sine, pulse, and triangle waveforms, or you can create custom waveforms.

Many actions have been developed for high throughput operations. These actions are generally named "Arb_Dl_xxx" where the "Dl" indicates the actions with the high throughput capability (see "Downloading and Executing Waveforms Using the "Arb_Dl_xx" Actions" below).

Note DO NOT use the "Arb_Dl_xxx" actions with the older "ArbSetSegment" and/or "ArbSetSequence" actions on the same Arb channel.

Note When selecting different downloaded waveforms, usually when there are major frequency differences, the first few cycles of a newly selected waveform may appear to be distorted. The reason is that the previous waveform must complete, but the new "clock adjust" is already in effect, which may change the timing of the last cycles of the previous waveform.

The following explains the methods used. For each method, there is a testplan available it. The testplans are located in the following directory:

C:\Program Files\Agilent\TS-5400 System Software\testplan\examples

In all testplans, the Arb connects to the Analog Bus ABus1. To view the output, connect an oscilloscope to the ABus 1 connector at the front of the system. Be sure to make the necessary connection to the Mass Interconnect/Express Connect so the interlock is shorted in order to use the system (see the appropriate documentation). Instead of an oscilloscope, you can also use an VXI Technology E1563A Digitizer to read the waveforms and display them using the appropriate actions.

Downloading and Executing Waveforms Using the "Arb_Dl_xx" Actions

Using the "Arb_Dl_xx" actions to download standard or user defined waveforms gives you the highest throughput capability. Using these actions, the waveforms are downloaded the first time the testplan is run. The waveforms remain in memory and can then be selected any time during testplan execution. The only time new waveforms are downloaded is when a testplan is modified.

Each downloaded waveform needs a unique name to keep track of the waveform. This name can then be used to call a particular waveform in any part of the testplan as many times as needed. Since both waveform data and other parameters, such as output voltage are also included with a name, the waveform will be output the same way every time it is selected.

Note When initializing the Arb for output using the "arbInitiate" action, the Arb outputs the last downloaded waveform, if "Arb_Select_Wave_By_Name" is not called before "arbInitiate". Use the waveform names in the 'waveform_name' parameters of the of "Arb_Dl_xxx" actions to select them.

A typical sequence to generate a waveform is as follows:

1. Setup Arb using the "arbConfOutControls" action
2. Send the setup data to the Arb using the "arbSet" action.
3. Download waveform data using the "Arb_Dl_xxx" action; use a unique waveform name in the 'waveform_name' parameter.
4. Initialize the Arb to output the waveform using the "arbInitiate" action.
5. Select the downloaded waveform using the "Arb_Select_Wave_By_Name" action using the waveform name in the "Arb_Dl_xxx" action.

Generating Standard Waveforms

The following shows how to download standard waveforms for output using the standard actions. For high throughput capability, use the “Arb_Dl_xx” actions.

Downloading a Standard Waveform

Use the “Arb_Dl_Std_Waveform” action to download the waveform and the “Arb_Select_Wave_By_Name” action to select the waveform. In this case, the same waveform name is referenced by both actions to select the waveform for output.

Using this method, a waveform can be selected on-the-fly to allow fast waveform access. This is because the waveform has previously been generated and stored in memory. All that is needed is to select the waveform to output it.

An example to download and then select a waveform is as follows. In the example, a waveform is generated and stored using a particular name (i.e., “Arbwave1”). Once this waveform is generated, it can then be selected in any part of the testplan.

> Generate the waveform using Arb_Dl_Std_Waveform with the waveform name = “Arbwave1” (parameter: ‘waveform_name’ = Arbwave1)

to select the waveform, set:

> waveform name of an Arb_Select_Wave_By_Name action = “Arbwave1” (parameter: ‘waveform_name’ = Arbwave1) and then execute that action.

Note The waveform names are NOT case sensitive.

Using the above actions allow you to download and store several different waveforms into the Arb that can be selected in different parts of the testplan. One way to do this is to download several different waveforms with different names at the beginning of the testplan and then select the different waveforms throughout the testplan.

An example to use different downloaded waveforms in different parts of a testplan is as follows:

```
Test1:    Arb_Dl_Std_Waveform: ‘waveform_name’ = sine
          Arb_Dl_Std_Waveform: ‘waveform_name’ = square
          Arb_Dl_Std_Waveform: ‘waveform_name’ = triangle
                •
                •
Test2:    Arb_Select_Wave_By_Name: ‘waveform_name’ = triangle
                •
                •
Test3:    Arb_Select_Wave_By_Name: ‘waveform_name’ = sine
                •
```

•

Test4: Arb_Select_Wave_By_Name: 'waveform_name' = square

Testplan '`arbdown_name.tpa`' illustrates how to download a 1 kHz sine wave, 2 kHz square wave, and 4 kHz triangle wave. A review of the testplan is below.

Test Group/Test/Action Name	**Description**
testgroup Arb download waveforms	Downloads standard waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download sine wave	Downloads the standard sine wave to the Arb.
Arb_Dl_Std_Waveform	Downloads a sine wave into the Arb. The 'function' parameter is set to "0" to download a sine wave. The 'waveform_name' parameter is set to "sine" to give the downloaded waveform that name. The 'frequency' parameter is set for a 1 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform. The 'waveform_name' parameter is set to
test download square wave	Downloads the standard pulse wave to the Arb.
Arb_Dl_Std_Waveform	Downloads a square wave into the Arb. The 'function' parameter is set to "1" to download a pulse. The 'waveform_name' parameter is set to "square" to give the downloaded waveform that name. The 'duty_cycle' parameter is set to "0.5" so the pulse function downloads a square wave. The 'frequency' parameter is set for a 2 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform.
test download triangle wave	Downloads the standard triangle wave to the Arb.
Arb_Dl_Std_Waveform	Downloads a triangle wave into the Arb. The 'function' parameter is set to "2" to download a triangle wave. The 'waveform_name' parameter is set to "triangle" to give the downloaded waveform that name. The 'frequency' parameter is set for a 4 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform.
test output waveforms	This test executes the downloaded waveforms.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform (i.e., "triangle").
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "sine"). Since this is the name of the 'waveform_name' parameter of the "Arb_Dl_Std_Waveform" action in "test download sine wave", a sine wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., sine wave).

Test Group/Test/Action Name	Description
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “square”). Since this is the name of the ‘waveform_name’ parameter of the “Arb_Dl_Std_Waveform” action in “test download sine wave”, a square wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., square wave).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “triangle”). Since this is the name of the ‘waveform_name’ parameter of the “Arb_Dl_Std_Waveform” action in “test download sine wave”, a triangle wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).

Outputting a Standard Waveform Immediately

The “Arb_Output_Std_Waveform” action can also be used to generate and output standard waveforms. This action enables the Arb to output a sine, pulse, or triangle wave. Unlike the “Arb_Dl_Waveform” action, this action does not require “arbIntiate” to enable it to output a waveform, however, it also has the high throughput capability of the “Arb_Dl_Waveform” action. “Arb_Output_Std_Waveform” can be used in any part of the testplan.

Testplan ‘`arboutputstd.tpa`’ illustrates how to output a 1 kHz sine wave, 2 kHz square wave, and a 4 kHz triangle wave. A review of the testplan is below.

Test Group/Test/Action Name	Description
testgroup Arb output std waveforms	Outputs standard waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test output sine	Output sine wave
Arb_Output_Std_Waveform	Immediately outputs a sine wave; ‘function’ parameter is set to “0”.
DialogOkay	This optional action is used here to view the current waveform (i.e., sine wave).
test output square	Output square wave
Arb_Output_Std_Waveform	Immediately outputs a square wave; ‘function’ parameter is set to “1” (pule) and ‘duty_cycle’ parameter is set to “0.5” for the “pulse” function to output the square wave.
DialogOkay	This optional action is used here to view the current waveform (i.e., square wave).
test output triangle	Output triangle wave
Arb_Output_Std_Waveform	Immediately outputs a triangle wave; ‘function’ parameter is set to “2”.
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).

Downloading a Standard Phase Shifted or Burst Modulated Waveform

Use the “Arb_Dl_Ext_Waveform” action for phase shifted and burst

modulated waveforms. This action has the following functionality:

1. Square wave generation
2. Positive and negative ramp generation
3. Positive and negative pulse generation
4. Burst modulation
5. Phase shift
6. Pulse width for positive and negative pulses, and burst pulses.

To correctly generate a waveform, it is important to understand how to set some of this action's different parameters. These are explained as follows:

frequency - Specifies the output frequency if the waveform is NOT a "burst" waveform ('F_burst' parameter set to "0"). If "burst" is used, this is the frequency of the burst wave, as shown in Figure 4-4. (Note: "burst" can not be used in the positive or negative pulse functions; i.e., functions 5 or 6, respectively.)

Tpulse_width - Sets the pulse width of the positive or negative pulses function (for functions 5 and 6, respectively) or sets the width, in seconds, of the burst wave, as shown in Figure 4-4. If used with "burst", the pulse width must be less than 100 mS.

F_burst - If set to "0", no "burst" is generated. Otherwise, this sets the waveform frequency that is to be modulated by the burst wave (the burst wave frequency is set by the 'frequency' parameter), as shown in Figure 4-4.

phase_shift - Specifies the amount of phase shift in a waveform in degrees, relative to a non-phase shifted version of itself.

force_seg_count - If set to "0", the frequency accuracy is generated using the 2 µS clock cycles plus any number of 100 nS clock cycles needed to make the frequency accurate. This is illustrated in the following formula:

segments = (1 / frequency) / (2 µS + (100 nS * clockadj))

where 'clockadj' is used to add 1 or more 100 nS ticks to the 2 µS clock.

If this parameter is set to "1" (or "-1"), the number of segments closest to the frequency value will be used, without using 'clockadj'. Only the 2 µS clock is used without the 100 nS clock. This is illustrated in the following formula:

segments = 1 / (frequency * 2 µS)

Normally, 'force_seg_count' is used (set to "1" or "-1") if you wish to have the segment counts of two waveforms be a multiple of each other to guarantee phase lock. However, this method uses more segment

memory since the number of segments determines the frequency.

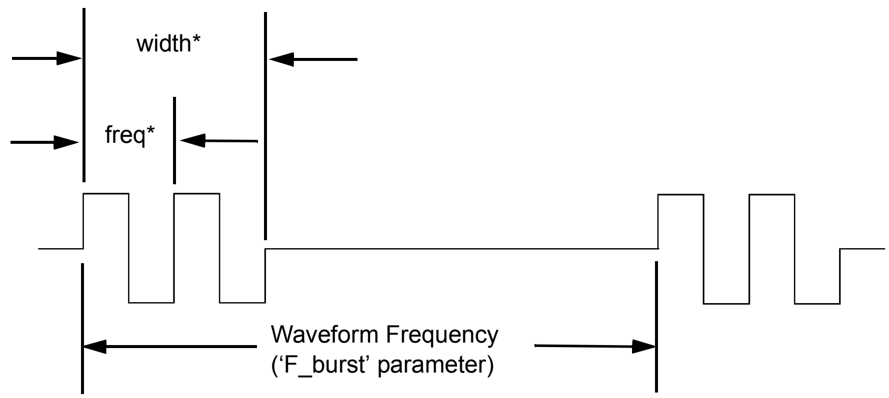


Figure 4-4. Typical Burst Waveform

Testplan '`arbburst.tpa`' illustrates how to generate a burst modulated waveform, with waveform frequency set to 30 kHz, burst frequency set to 1 kHz, and burst pulse width set to 0.0001 seconds. A review of the testplan is below.

Test Group/Test/Action Name	Description
testgroup Arb burst waveform	Outputs a burst waveform.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download burst	Setup arb to download a burst waveform.
Arb_Dl_Ext_Waveform	Setup a burst waveform with waveform frequency set to 30 kHz ('F_burst' parameter), burst frequency of 1 kHz ('frequency' parameter), and a burst pulse width of 100 µS ('Tpulse_width' parameter).
test output waveform	Selects and outputs a burst waveform by name.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "wave1").
DialogOkay	This optional action is used here to view the current waveform (i.e., forward and reverse sweep).

Downloading a Swept Sine Waveform

The “Arb_Dl_Swept_Sine” and “Arb_Dl_Swept_Sine_Ex” actions perform frequency sweeps of sine waves. The “Arb_Dl_Swept_Sine” action only sweeps in the forward direction. The “Arb_Dl_Swept_Sine_Ex” action can sweep in both forward and reverse directions with a selectable delay between sweeps (i.e., last frequency remain for the selected time).

Testplan ‘`arbsweep.tpa`’ illustrates how to output a sine wave sweep using both actions. In one test, the sine wave sweeps from 1 kHz to 2 kHz with a sweep time of 0.5 seconds. In another test, the sine wave sweeps from 1 kHz to 2 kHz in both forward and backwards directions with a sweep time of 0.2 seconds. In the The latter test, the last frequency is held for 0.5 seconds before the next sweeps is initiated. A review of the testplan is below.

Note The number of segments needed to generate the swept sine waves depend on the frequency span between the start and end frequencies, and the sweep time. Large spans and/or sweep times could exceed the maximum allowable segments.

Test Group/Test/Action Name	Description
testgroup Arb sine sweeps	Sweeps Arb sine waves.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb’s output circuitry.
arbSet	Sends setup data to Arb.
test setup forward sweep	Sweeps a sine wave in the forward direction.
Arb_Dl_Swept_Sine	Downloads a sine wave for a forward sweep from 1 kHz to 2 kHz for a time of 0.5 seconds.
test setup forward & reverse sweeps	Sweeps a sine wave in both forward and reverse direction.
Arb_Dl_Swept_Sine_Ex	Downloads a sine wave for a forward and reverse sweep from 1 kHz to 2 kHz for a time of 0.2 seconds with a delay of 0.5 seconds between sweeps.
test execute sweeps	Execute both sweeps.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform (i.e., forward and reverse sweep).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”).
DialogOkay	This optional action is used here to view the current waveform (i.e., forward sweep).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave2”).
DialogOkay	This optional action is used here to view the current waveform (i.e., forward and reverse sweep).

Note Use the Watch Window to determine the amount of segment memory used by the swept sine waveforms.

Generating Custom Waveforms

Custom waveforms can be generated using the "Arb_Dl_Waveform_Data" action and or the "Arb_Dl_Custom_Waveform" actions.

Downloading a Data Type Custom Waveform

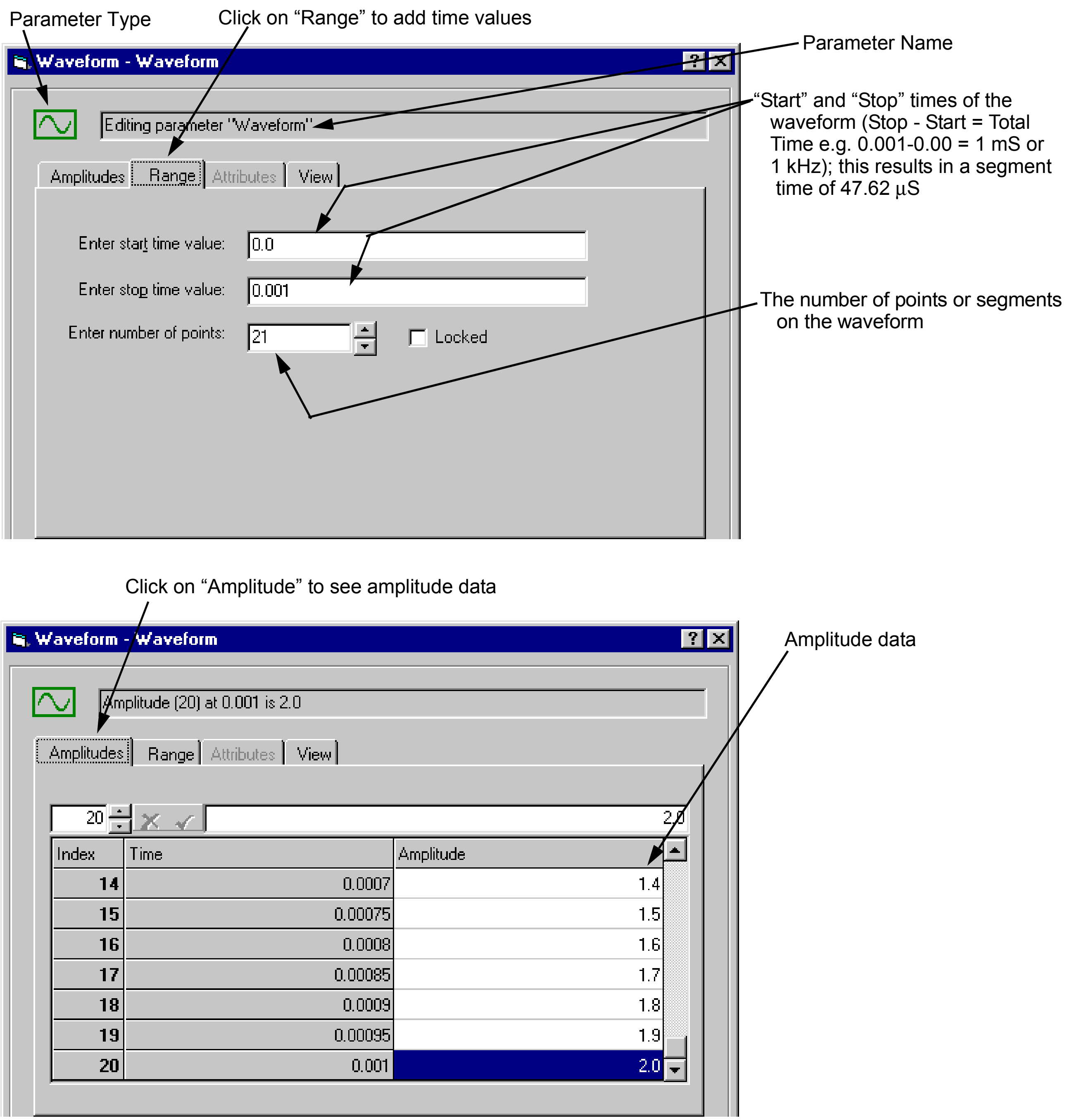
The "Arb_Dl_Waveform_Data" action uses a 'Waveform' type array variable called "Waveform" to generate a custom waveform. The variable stores both the segments of the waveform (i.e., amplitude) and the time all segments are to be executed (i.e., waveform frequency).

Figure 4-5 shows the data used to generate the waveform shown in the bottom part of the figure. The segment data of the waveform is stored in the array part of the "Waveform" parameter. (You can view the resultant waveform by clicking on the "View Waveform" button, as shown in Figure 4-5.)

How to download custom data type waveforms is shown in testplan '`arbcustom_data.tpa`'. The waveform is a 1 kHz (1 mS), 2 V peak ramp, consisting of 21 segments. A review of the testplan is below.

Test Group/Test/Action Name	**Description**
testgroup download custom waveform	Downloads a user generated data type custom waveform.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download custom waveforms	Downloads custom waveforms to Arb.
Arb_Dl_Waveform_Data	Downloads data type waveform into Arb. The waveform data is the data stored into the 'Waveform' variable (see Figure 4-5 for typical waveform data). The 'waveform_name' parameter is set to "wave1" to give the downloaded waveform that name. The maximum value in the 'Waveform' array variable is "2.0000000000" which makes the peak Waveform voltage 2.0 V. The Star and /Stop values in the 'Waveform' array are set to "0.0000000000" and ".0010000000", respectively. This outputs a 1 kHz waveform (Stop - Start = .0010000000 - 0.0000000000 = 1 mS or 1 kHz).
test output custom waveform	This test executes the downloaded waveform. Although the Arb outputs the waveform from the previous test (after 'arbinitiate' in "test download custom waveform"), this test is to show that the waveform can be selected in any part of the testplan. The Arb normally outputs the last downloaded waveform, whether a custom waveform or a standard waveform, but can output a different waveform in any part of the testplan using the action in this test.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform.

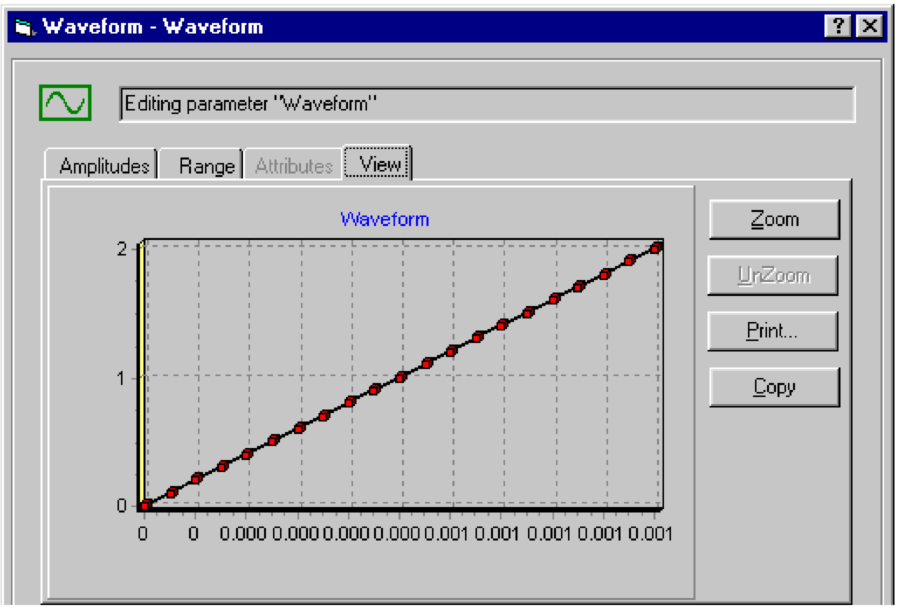
Test Group/Test/Action Name	Description
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”) This is the same name as the name in the ‘waveform_name’ parameter of the “Arb_Dl_Waveform_Data” action.
DialogOkay	This optional action is used here to view waveform.



Click on “View Waveform” to view resultant waveform (see next figure)

Figure 4-5. Generating a Data Type Waveform

The values in this example produces a 2 Vpeak, 1 kHz ramp, as shown below; the waveform was generated using the following values



0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
and Stop / Start values of
0.000 / 0.001

Note: The Arb output looks more “staircased” on an oscilloscpe due to the 47.62 μS/segment time.

Figure 4-6. Arb Waveform

Downloading User Defined Waveforms

This “Arb_Dl_Custom_Waveform” action uses several double and single dimensioned array type parameters to set the voltage segments, sequences, markers, dwell, next sequences, and other values to download and generate user defined custom waveforms.

Figure 4-3 shows a typical custom waveform. The figure shows a triangle waveform that shows the relationship between the segments, sequence, and segment execution time.

To generate a waveform, it is important to understand how to set the different parameters of the action. This is explained as follows:

voltage - Specifies the voltage value of the waveform segments for one or more sequences (voltage range is ±16.4 V). This is a double dimensioned array where the column is the sequence number and the rows are the segment numbers. Figure 4-7 shows a typical example.

marker - Specifies a marker on the selected segment in a valid sequence. The marker pulse will only be output if the sequence is non zero. The marker signal always appears on the Arb’s front panel BNC connector. It can also be routed to the VXI TTL backplane. Figure 4-8 shows a typical example on how to specify markers.

nr_sequence - Specifies the number of sequences to be downloaded. This value must be “1” or above, or no waveform will be downloaded. If more than one sequence is available and the parameter is set to “1”,

only the waveform using the first sequence is downloaded.

next_sequence - Specifies which sequence is to be downloaded next. An example is in Figure 4-9.

sequence_repeat - Specifies if the sequence is to be repeated (i.e., value is set to “1” or more), as shown in Figure 4-10.

nr_segments - Specifies the number of segments for each sequence. If the number of specified segments is lower than the number of segments entered in the corresponding sequence of the ‘voltage’ parameter, only the number of segments specified here will be downloaded. Figure 4-11 shows a typical example.

segment_dwell - Specifies for how many 2 µS clock cycles each segment of a sequence is to be output. Normally, each segment is output in a single 2 µS clock cycle (if the parameter is set to “0”), which is a dwell count of “0”. Setting the ‘segment_dwell’ to a number above zero, the clock cycles will increase by that new number plus a single clock cycle.

Since the number of segments and the duration time of the segments determines the waveform frequency, each additional number added to ‘segment_dwell’ lowers the waveform frequency.

For example, if the dwell count is 0, for a 20 segment waveform, the total time the segments are output is:

$$20 * 2\ \mu S = 40\ \mu S = \text{a frequency of } 25\ \text{kHz}$$

Changing the dwell count to 49, changes the time and resultant waveform frequency to:

$$20 * ((49 * 2\ \mu S) + 2\ \mu S) = 20 * ((98\ \mu S) + 2\ \mu S) = 20 * 100\ \mu S = 0.002\ S = \text{a frequency of } 500\ \text{Hz}.$$

An example of using the ‘segment_dwell’ parameter is in Figure 4-12. In the figure, the dwell count is 49 for two 20 segment sequences which gives a total waveform frequency of 250 Hz.

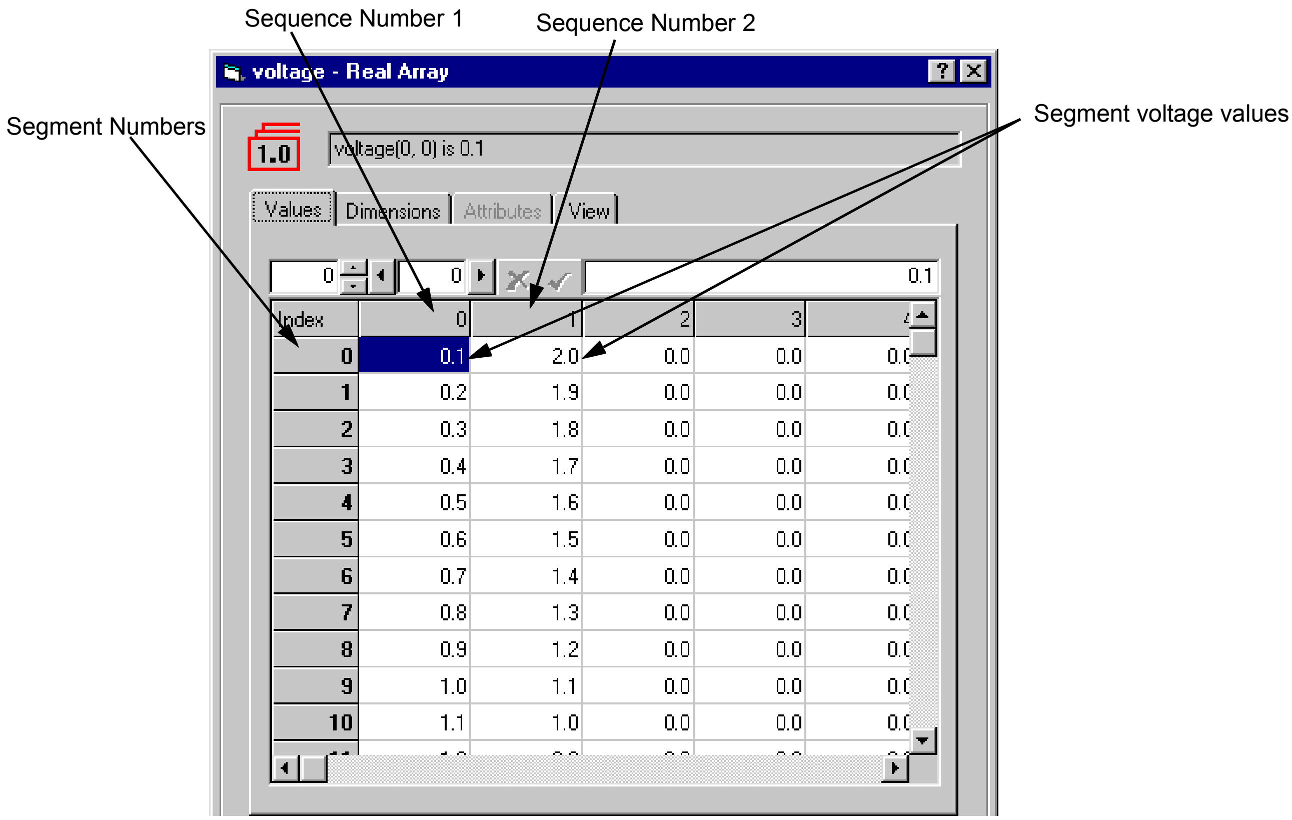


Figure 4-7. Segment/Sequence Array Values

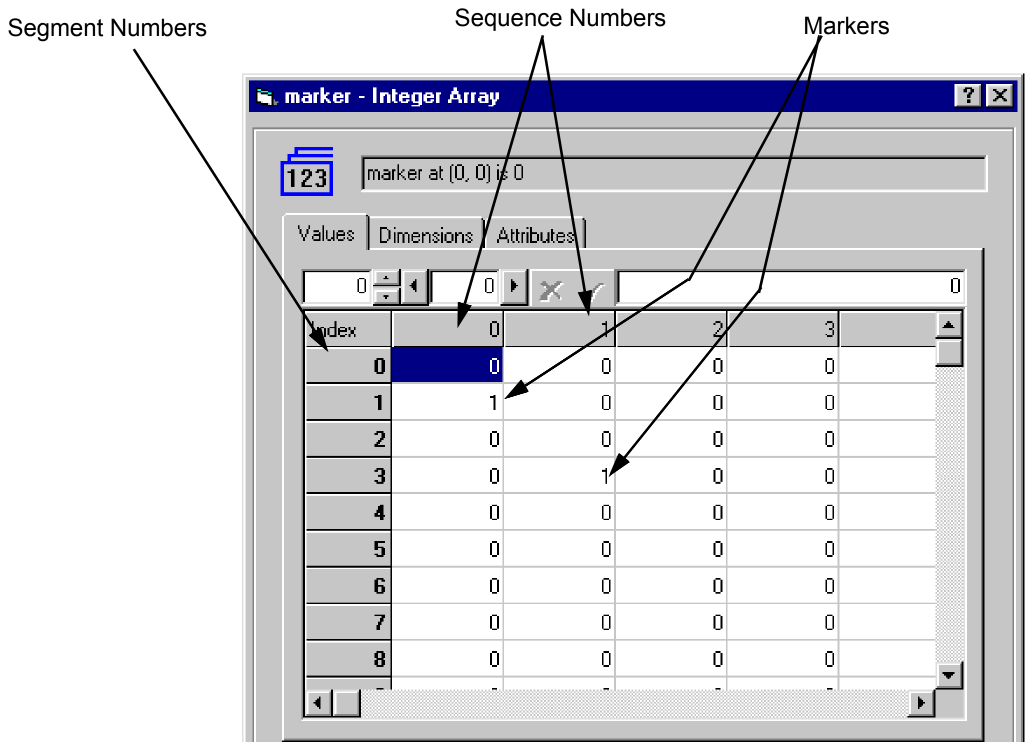


Figure 4-8. Typical Marker Values

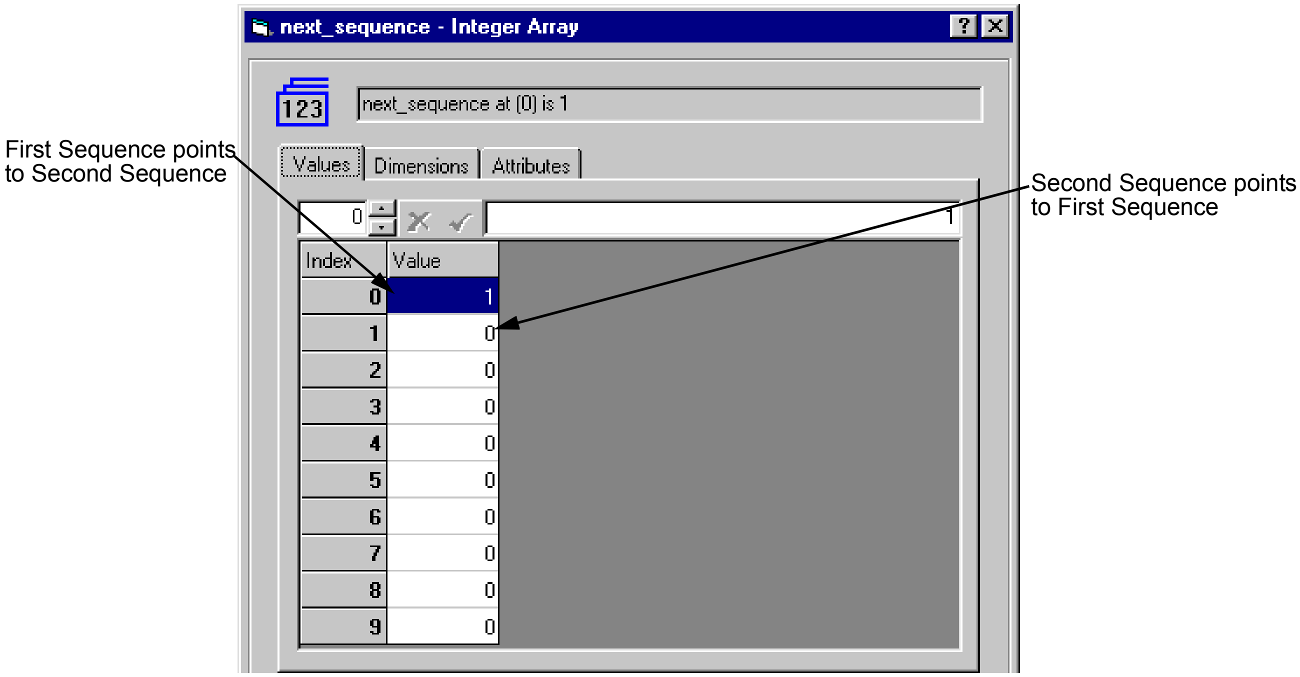


Figure 4-9. Typical Next Sequence Values

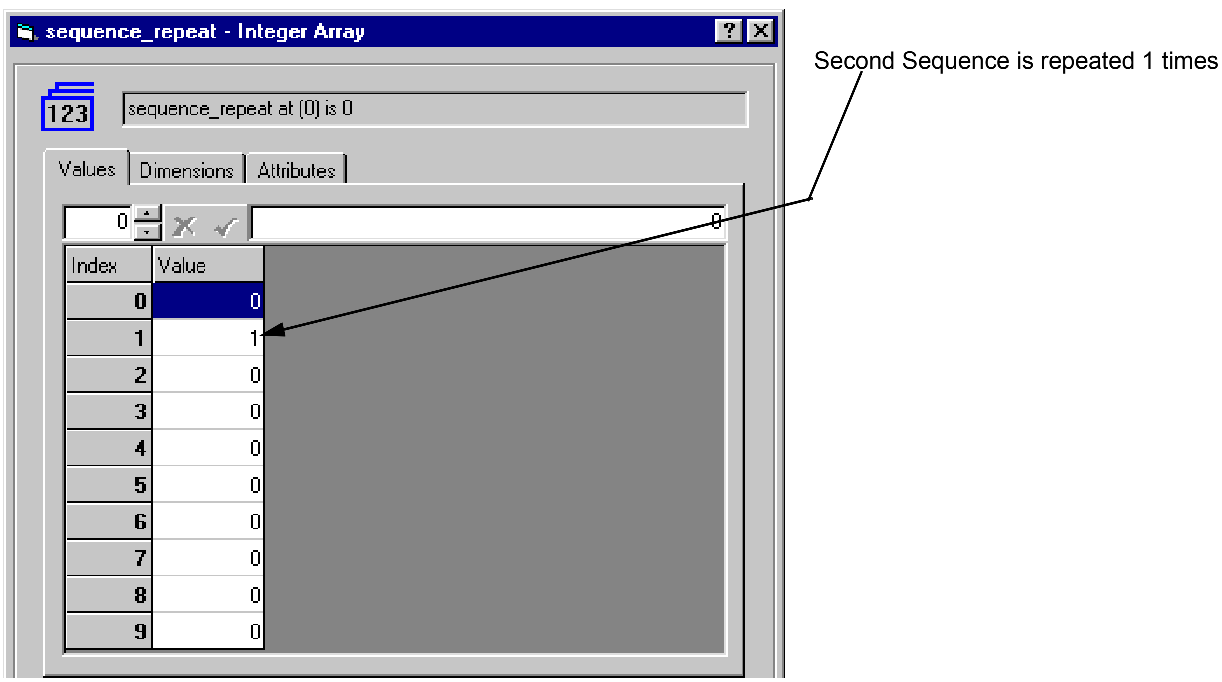


Figure 4-10. Typical Repeat Sequence Value

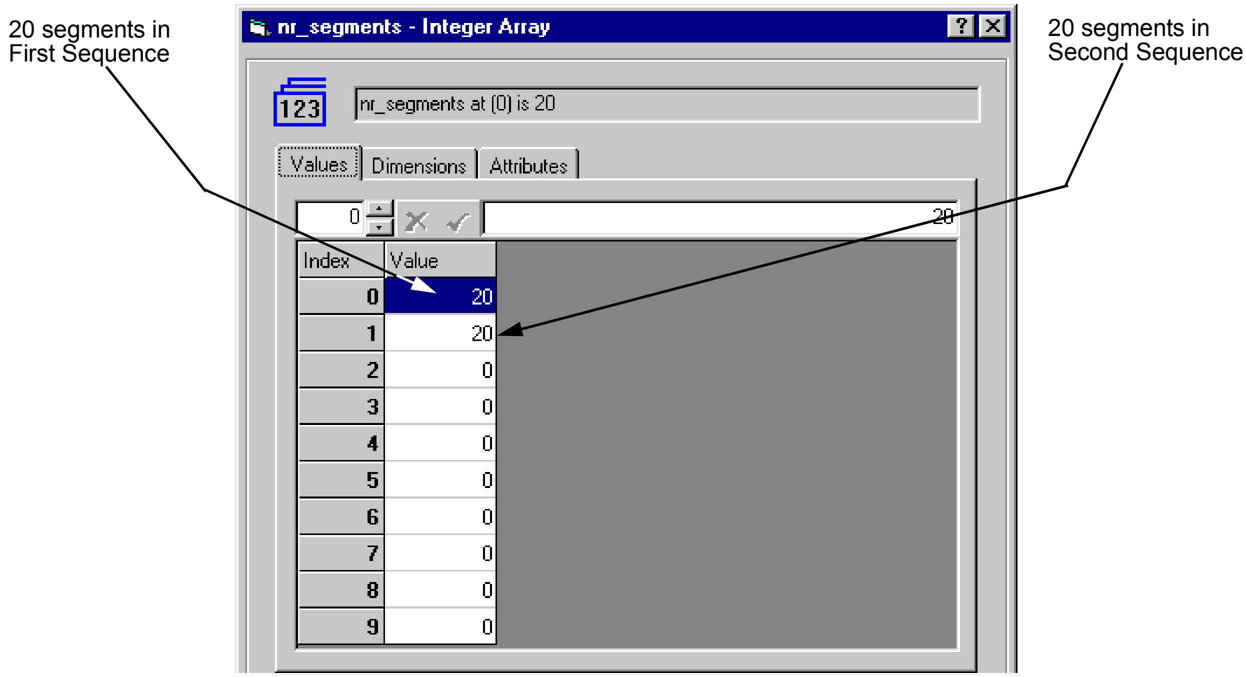
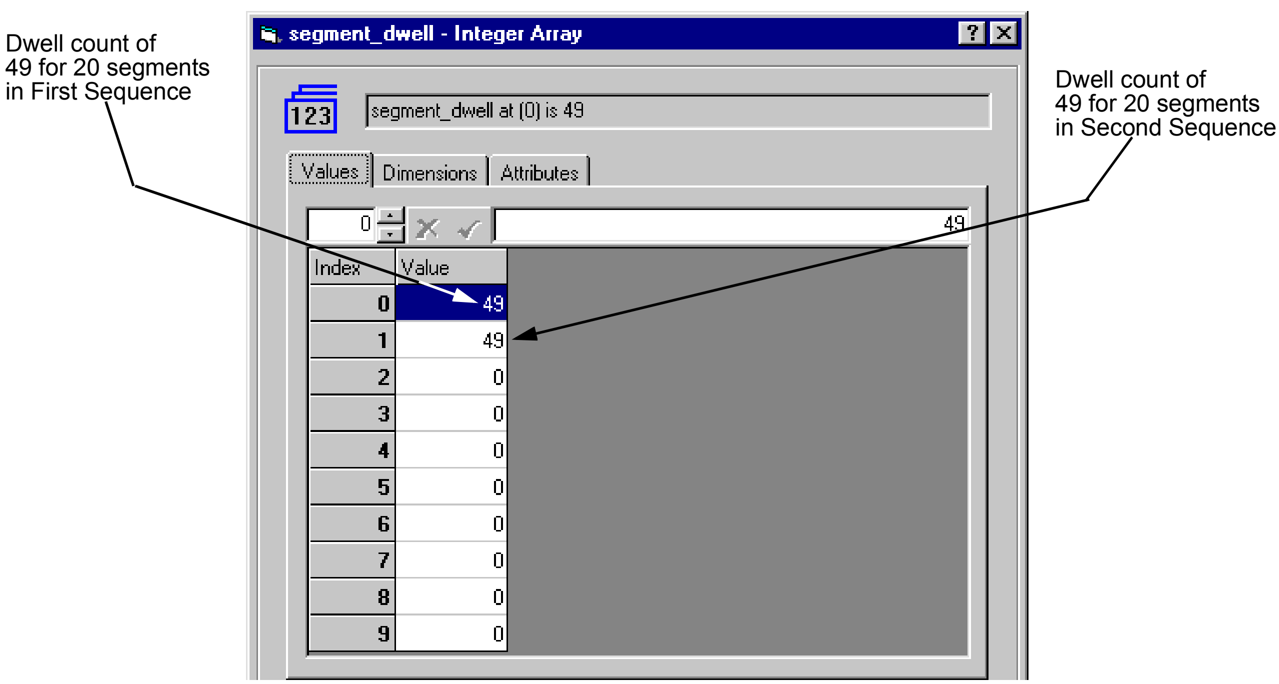


Figure 4-11. Typical Segment Number Value



Waveform time for both sequences is: total of (segments * clock time for each sequence) =
20 * ((49 * 0.000002) + 0.000002) = 20 * (0.000098 + 0.000002) = 20 * 0.0001 = 0.002 per sequence

Since both sequence time is the same, the total time = 2 * 0.002 = 0.004 = 250 Hz

Figure 4-12. Typical Segment Number Value

How to download custom waveforms is shown in testplan ‘`arbcustom_user.tpa`’. A review of the testplan is as follows:

Test Group/Test/Action Name	Description
testgroup download custom waveform	Downloads user generated custom waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb’s output circuitry.
arbSet	Sends setup data to Arb.
test download custom waveforms	Downloads the custom waveforms to Arb.
Arb_Dl_Custom_Waveform	Downloads two ramps (one positive the other negative going) to the Arb to output a triangle waveform. The number of segments is 20 for each waveform. The waveforms require 2 sequences. The ‘segment_dwell’ is set to 49 for both waveforms to output at 250 Hz.
test execute waveforms	Executes custom waveforms.
arbInitiate	Initialize the Arb to output the waveforms. Start outputting the waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”) This is the same name as the name in the ‘waveform_name’ parameter of the “Arb_Dl_Custom_Waveform” action.
DialogOkay	This optional action is used here to view waveform.

Generating Custom Waveforms From a Data File

This is similar to generating a data type custom waveform using the “Arb_Dl_Waveform_Data” action, except in this case the waveform data comes from a data file using the “WaveformDataRead” action. The following shows how to download custom waveform data from a file and other pertinent information.

Downloading from a File

The following shows how the actions are used:

1. The “WaveformDataRead” action reads the data from a file and stores it into its ‘Waveform” variable (at first run of the testplan only).
2. The data is then stored into the ‘Waveform’ variable of the “Arb_Dl_Waveform_Data” action.
3. The data is downloaded into the Arb and output whenever the “Arb_Select_Wave_By_Name” action selects the appropriate waveform name.

To perform the correct operation, the following must take place:

- The “WaveformDataRead” action must be referenced to the data file using the file name. Figure 4-13 shows how the file is referenced.

  Normally, “WaveformDataRead” action reads the data file when the testplan is first run (i.e., the System RunCount = 0). After that, the data file will not be read.

- The waveform time/frequency must be set by the ‘SampleTime’ parameter in the “WaveformDataRead” action. Figure 4-13 shows how to set the waveform time/frequency using the ‘SampleTime’ parameter.

- The ‘Waveform’ variable in the “Arb_Dl_Waveform_Data” action must be referenced to the ‘Waveform’ variable in the “WaveformDataRead” action using symbols. Figure 4-14 shows how to create a symbol in the “WaveformDataRead” action and Figure 4-15 shows how to reference to the symbol in the “Arb-Dl_Waveform_Data” action.

- The data in the file must be in such a format that each line is a data point or segment of the waveforms. The following is a typical example on how data is stored into the file:

  0.0000000000
  0.1000000000
  0.2000000000
  •
  •
  •
  0.5000000000

Note A data file using this format can be generated by an E1563A Digitizer using an appropriate action or the using the “WaveformDataWrite” or other supplied standard action.

How to download a custom data type waveform is shown in testplan ‘`arbcustom_file.tpa`’. The downloaded waveform is a 1 kHz (1.0 mS), 2 V peak ramp, consisting of 21 segments. A review of the testplan is as follows:.

Test Group/Test/Action Name	Description
testgroup Arb waveform from file	Downloads a waveform from a file.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb’s output circuitry.
arbSet	Sends setup data to Arb.
test get and download data	Gets download waveform data from file and downloads the data to the Arb.
WaveformDataRead	Reads data from file and store it in an array variable.
Arb_Dl_Waveform_Data	Download the data into the Arb that was read from the file to generate a data type waveform. The ‘waveform_name’ parameter is set to “wave1” to give the downloaded waveform that name. The waveform time or frequency is set to 1 kHz using the ‘SampleTime’ parameter value in the “WaveformDataRead” action where frequency = 1 / (sample time * number of samples) = 1 / (0.000047619 * 21 = 0.001 S = 1 kHz
test output custom waveform	Executes the downloaded waveform.
arbInitiate	Initialize the Arb to output the waveforms. Start outputting the waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”) This is the same name as the name in the ‘waveform_name’ parameter of the “Arb_Dl_Waveform_Data” action.
DialogOkay	This optional action is used here to view waveform.

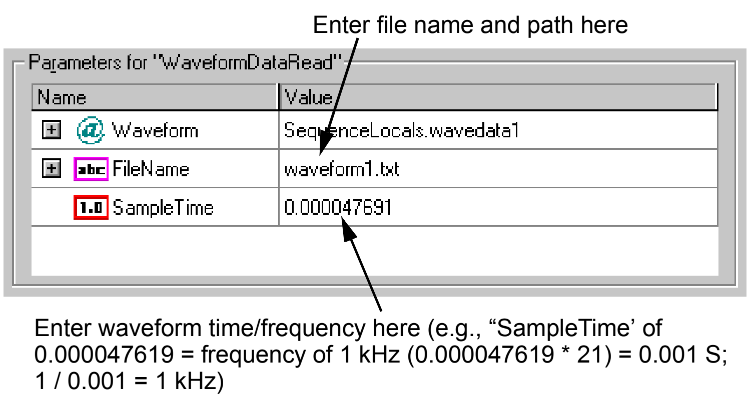


Figure 4-13. Entering File Name and Sample Time

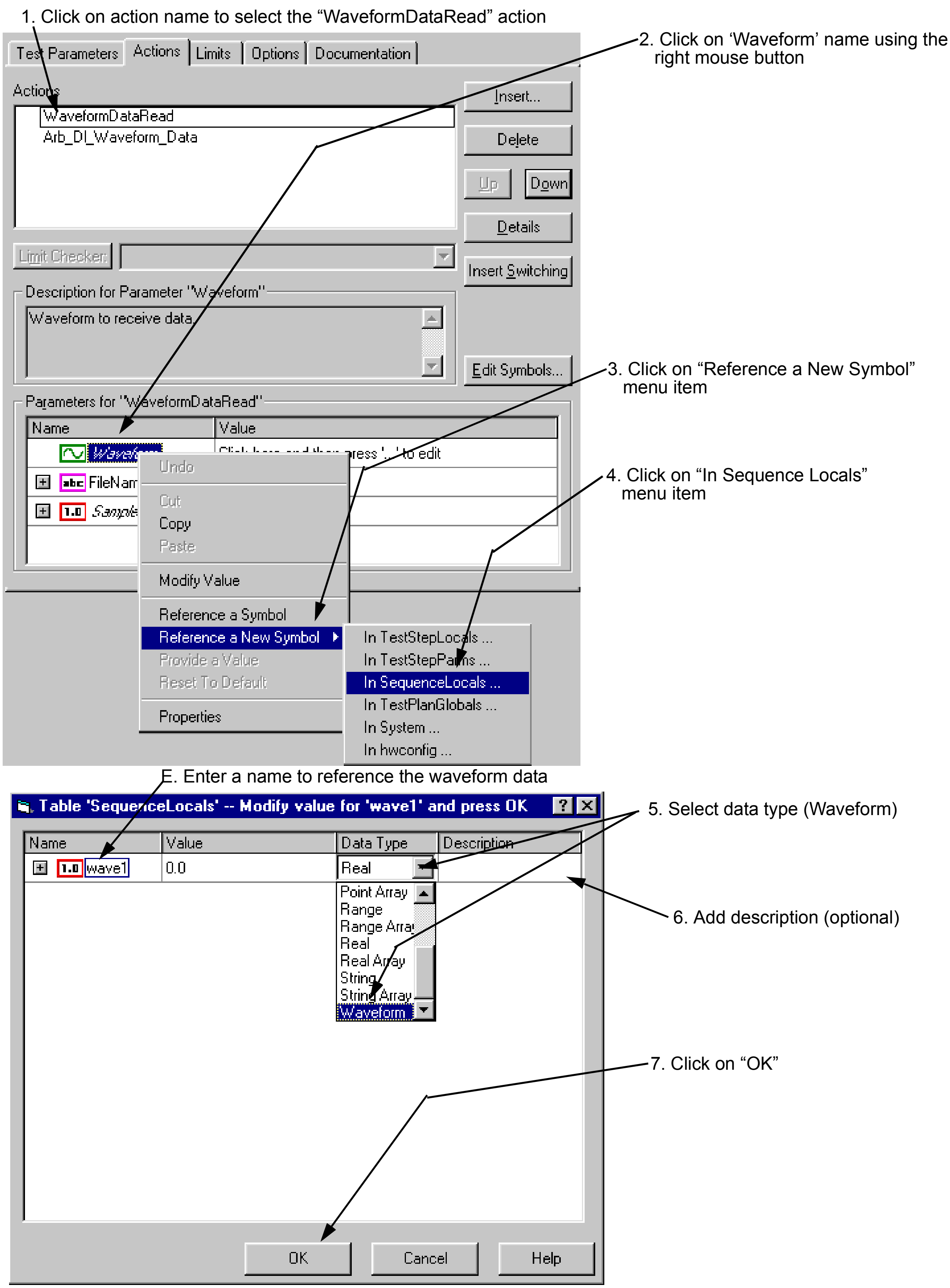
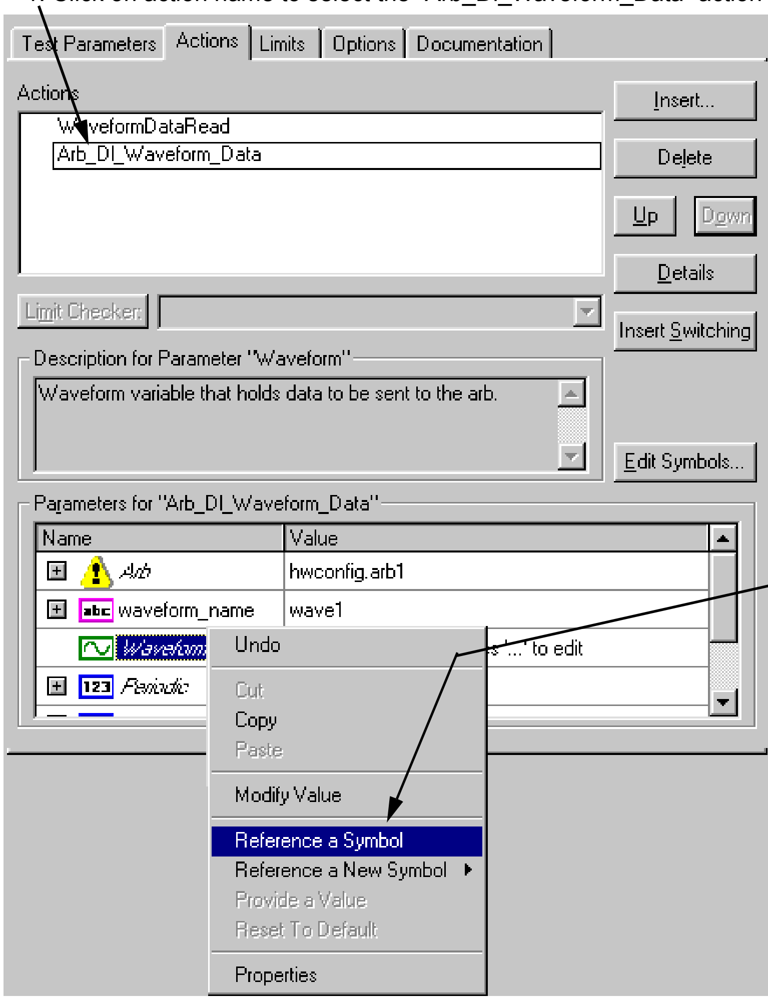


Figure 4-14. Create a Symbol for the "WaveformDataRead" Action

1. Click on action name to select the “Arb_DI_Waveform_Data” action

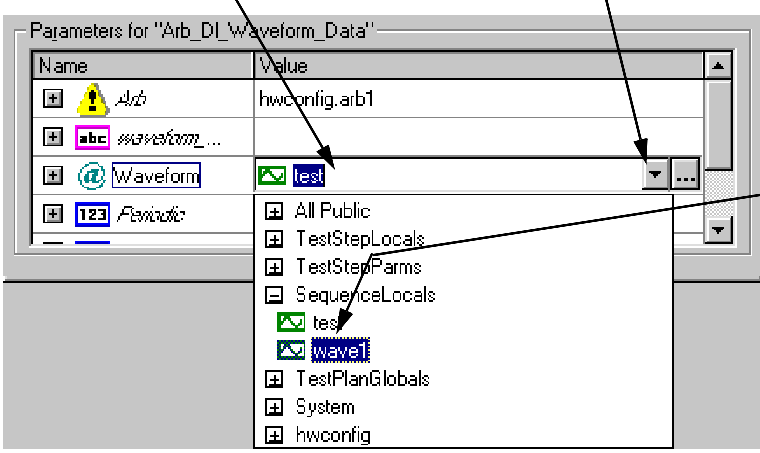


2. Click on “Reference a Symbol” menu item

3. The program automatically selects a symbol. If only one symbol is stored, it selects that symbol. If more than one symbol is stores, you must select the symbol, as shown in steps D to F

4. Click on parameter value to select the parameter

5. Click on the arrow to select the symbol (you can also click on the three dots to select it, but a different procedure is used)



6. Click on the appropriate symbol name to select it

Figure 4-15. Reference a Symbol in the “Arb_DI_Waveform_Data” Action

Creating a Data File

The data file is an ASCI file that contains waveform data. The data can be entered and added manually where each segment data point is a line in the file. You can also use the "WaveformDataWrite" action to store the data into the file. For example, you can use the action to store a waveform read by the E1563A Analog to Digital Converter into a file which then can be downloaded to the Arb and output.

Memory Requirements for Downloading Waveforms

In order for the Arb to achieve high throughput, it downloads all waveforms that are to be used in the testplan into memory. A waveform can then be output in any part of the testplan.

Although the Arb can download many waveforms, there is a limit on how many waveforms can be stored. This depends on the size of the waveforms. Normally, the Arb has sufficient memory for all the waveforms in a testplan. However, for testplans that use a very large number of waveforms (or several long waveforms), the available memory may be insufficient to hold them all. The Arb driver will then generate an error; normally an "Segment count too big during download" error (see "Agilent E6173A Arbitrary Waveform Generator Error Messages" below).

In case of a memory overflow, the only way for the Arb to receive new waveforms is to remove the current waveforms from memory. To do this, use the "arbReleaseWaveforms" action. This action will then remove all waveforms downloaded for the selected channel.

Note It it becomes necessary to use the "arbReleaseWaveforms" action, place the action both at the beginning of the testplan and at the point where more waveforms need to be added.

Agilent E6173A Arbitrary Waveform Generator Error Messages

The following lists the error numbers in hex and corresponding error messages the Arb can generate. Use the messages to troubleshoot the testplan or Arb, if needed.

Table 4-1. Agilent E6173A Error Codes and Descriptions

Error Number	Description
21	Unimplemented function call
22	Timeout waiting for arbStop
23	Illegal segment voltage
24	Segment count too big during download
25	Segment range error (setSegment)
26	Sequence range error (setSequence)
27	Illegal trigger input
28	Not channel 1 or 2
29	Sequence function without a segment fn
2A	Segment funct. without a sequence fn

Table 4-1. Agilent E6173A Error Codes and Descriptions

Error Number	Description
2B	Couldn't find specified download ID
2C	Sequence count too big during download

Chapter 5

Action Summary

This chapter summarizes the TestExec SL actions supplied with the Agilent TS-5400 System. The action directory path is:

C:\Program Files\Agilent\TS-5400 System Software\actions

Actions are located in the following sub-directories:

Sub-Directory	Action Type
arb	Arbitrary Waveform Generator Actions
counter	Counter Actions
dac	D/A Converter Actions
dgn	Diagnostics Actions
digitizer	Digitizer Actions
dio	Digital I/O Actions
dmm	Digital Multimeter Actions
event	Event Detector Actions
generic	Miscellaneous Actions
mcm	Measurement Control Module Actions
power	Power Supply Actions
serial	Serial Interface Actions
SerialProtocol	Automotive Serial Protocol Actions
switch	SLU and Switching Actions

Note This chapter contains action summaries. The actions are documented in detail in the TS5400 online help. This help is available from the Help menu in TestExec SL.

Table 5-1. arb (Arbitrary Waveform Generator) Actions

Agt33220GetBurstProperties	Returns the values of the IAgilent33220Burst interface properties.
Agt33220GetOutputProperties	Returns the values of the IAgilent33220Output interface properties.
Agt33220MsgReceive	Receives SCPI commands via the IAgilent33220System interface.*
Agt33220MsgSend	Sends SCPI commands via the IAgilent33220System interface.*
Agt33220MsgSendReceive	Receives SCPI commands via the IAgilent33220System interface.*
Agt33220OutputArbWaveform	Downloads an arbitrary waveform and enables the output.
Agt33220OutputStdWaveform	High level action to generate a basic waveform.

Table 5-1. arb (Arbitrary Waveform Generator) Actions

Agt33220PutBurstProperties	Sets the values of the IAgilent33220Burst interface properties.
Agt33220PutOutputProperties	Sets the values of the IAgilent33220Output interface properties.
arbConfOutControls	Configures the Agilent E6173 Arb's output circuitry.
arbConfSync	Enable/disable synchronous mode on a given arb channel pair.
arbConfTimebase	Select internal 10MHz clock or an external clock supplied through the front panel BNC.
arbConfTimeout	Sets the I/O timeout value (in milliseconds) for communication with the Agilent E6173 Arb.
arbConfTrigIn	Set input triggering parameters.
arbConfTrigOut	Set trigger parameters for output.
arbInitiate	Starts the arb sequencer to output the waveform.
arbIsSet	Program arb - transfers all arb settings specified by the arbConf.
arbReleaseWaveforms	Causes the Agilent E6173 handler and driver to "forget" about all previously downloaded waveforms.
arbReset	Causes a hard reset of the arb hardware.
arbSet	Program arb - transfers all arb settings specified by the arbConf.
arbSetClkAdjust	Change a channel's clock timing on the fly.
arbSetSegment	Not recommended for use in new testplans (use Arb_Dl_Custom_Waveform).
arbSetSequence	Not recommended for use in new testplans (use Arb_Dl_Custom_Waveform).
arbSetWaveform	Not recommended for use in new testplans.
arbStop	Stops waveform output, at end of current sequence, on a given channel.
Arb_Disable_Download	Not recommended for use in new testplans.
Arb_Dl_Custom_Waveform	Downloads a custom, user defined waveform consisting of up to 10 sequences with 50 segments per sequence to the Agilent E6173 Arb.
Arb_DL_Datafile	Creates a waveform from a data file.
Arb_Dl_Ext_Waveform	Downloads a sine, square, or triangle waveform of specified frequency, amplitude, offset and phase (with optional burst modulation) to the Agilent E6173 Arb.
Arb_Dl_Std_Waveform	Downloads a sine, square/pulse or triangle waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_Dl_Swept_Sine	Downloads a basic swept sine waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_Dl_Swept_Sine_Ex	Downloads a swept sine waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).

Table 5-1. arb (Arbitrary Waveform Generator) Actions

Arb_Dl_Waveform_Data	Downloads a waveform data type to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_Download	Not recommended for use in new testplans (use "Arb_Dl_Std_Waveform" or "Arb_Dl_Ext_Waveform").
Arb_Download_Deluxe	Not recommended for use in new testplans (Use "Arb_Dl_Ext_Waveform" or "Arb_Dl_Custom_Waveform").
Arb_Download_Waveform	Not recommended for use in new testplans (use "Arb_Dl_Waveform_Data").
Arb_Enable_Download	Not recommended for use in new testplans.
Arb_Output_Std_Waveform	Configures Agilent E6173 Arb to generate a sine, pulse, or triangle waveform of specified frequency, amplitude, and offset.
Arb_Select_Synced_Waveforms	Not recommended for use in new testplans (Use "Arb_Select_Synced_Waves_By_Name").
Arb_Select_Synced_Waves_By_Name	Allows selection of Agilent E6173 Arb waveforms (on the fly) of both channels referenced by name assigned in the "Arb_Dl_xxx" actions.
Arb_Select_Wave_By_Name	Allows the selection of Agilent E6173 Arb waveforms (on the fly) referenced by name assigned in the "Arb_Dl_xxx" actions.
Arb_Select_Waveform	Not recommended for use in new testplans (Use "Arb_Select_Wave_By_Name").
Arb_Source_MDA	High-Level Action: Programs the arb to act as the MDA source.
CreateArbWaveform	This program uses the arbitrary waveform function to create an array that defines a square wave pulse with a calculated rise time and fall time. The waveform consists of 4000 points.

Table 5-2. counter Actions

Low-Level Actions	
ctrConfFunction	Selects function, range, and resolution of the Agilent E1333 Counter.
ctrConfInControls	Configures the input circuit of the Agilent E1333 Counter.
ctrConfTrigIn	Sets the input trigger of the Agilent E1333 Counter.
ctrGetResults	Reads the Agilent E1333 Counter results.
ctrInitiate	Starts the previously configured Agilent E1333 Counter measurement.
ctrIsSet	Waits until the Agilent E1333 Counter is ready for measurement.
ctrReset	Resets all of the Agilent E1333 Counter's channels to the power-on state.
CtrSelfTest	Runs a selftest on the Agilent E1333 Universal Counter that is provided by its Plug and Play driver.
ctrSetReturnMode	This action allows time-outs on the specified Agilent E1333 Universal Counter to raise an exception when encountered.
ctrSetTimeout	Sets the Agilent E1333 Counter operation time out value (in milliseconds).
High-Level Actions	
ctrMeasureFrequency	An Agilent E1333 Counter action that measures frequency.

Table 5-2. counter Actions

ctrMeasurePeriod	An Agilent E1333 Counter action that measures period.
ctrMeasurePulseWidth	An Agilent E1333 Counter action that measures pulse widths.
ctrMeasureRatio	High level Agilent E1333 Counter Action to measure Ratio using the counter.
ctrMeasureTimeInterval	High level Agilent E1333 Counter action that measures time interval.
ctrMeasureTotalize	High level Agilent E1333 Counter action that starts a totalize measurement.

Table 5-3. dac (D/A Converter) Actions

dacConfSourceDCI	Configures the Agilent E1418 DAC to output the specified current (Agilent E1418 only).
dacConfSourceDCV	Configures the Agilent E1418 DAC to output the specified voltage (Agilent E1418 only).
dacConfTrigIn	Selects the Agilent E1418 DAC's trigger source (Agilent E1418 only) for all channels.
dacConnect	Immediately connects the output relays on the selected Agilen E1418 DAC (Agilent E1418 only).
dacDisconnect	Immediately disconnects the output relays on the selected Agilent E1418 DAC (Agilent E1418 only).
dacIsSet	Waits until the Agilent E1418/E1328 DAC is ready for output.
dacReset	Resets all channels of the Agilent E1418/E1328 DAC to the power-on state (0V output) and disconnects the output relay.
dacSelftest	Runs selftest on the Agilent E1418/E1328 DAC; returns a pass/fail indication.
dacSet	Sends current setup information to the Agilent E1418 DAC (Agilent E1418 only).
dacSetDCI	Sets the Agilent E1418/E1328 DAC to output the specified current; causes an immediate output.
dacSetDCV	Sets the Agilent E1418/E1328 DAC to output the specified voltage; causes an immediate output.
dacSetDCVSU	Program a DAC channel on the Agilent E6198 switch/load unit to a desired voltage.
dacSetGainOffsetSU	Change the gain/offset values used to set the DAC voltage in the switch/load unit.
dacSetGainSU	Change the gain value used to set the DAC voltage in the switch/load unit.
dacSetOffsetSU	Change the offset value used to set the DAC voltage in the switch/load unit.
dacSetPSVSU	Program a DAC channel voltage on the Agilent E6198 switch unit using the gain value in the module parameter block.
dacSetTimeout	Sets the Agilent E1328 or E1418 Digital to Analog Converter (DAC) operation time out value (in milliseconds).

Table 5-3. dac (D/A Converter) Actions

High-Level Actions	
DAC: 16 Chan (Individual)	This action programs each channel of the 16 channel Agilent E1418 DAC either as a voltage or current source.
DAC: 16 Channels	This action programs all channels of the 16 channel Agilent E1418 DAC either as a voltage or current source.

Table 5-4. dgn (Diagnostic) Actions

Ag3499SlotCheck	Verifies that the 3499 contains the N2261, N2262, and N2264 cards in one of the 5 slots
ArbAdjust	Measures Voltage/Current (V/I) gains using the MCM and DMM on external Arb connections to the system.
AutoAdjust	Measures all Voltage/Current (V/I) gains and offsets using the MCM and DMM.
dgnConnectString	Connects the switch path referenced by the parameter 'pathstring'. This action is helpful when the path you wish to connect may be a variable or a symbol table entry.
dgnDisConnect	Disconnects the switch path referenced by the parameter 'path'. This action is helpful when the path you wish to disconnect may be a variable or a symbol table entry.
EFTMuxTestDebugPorts	This action tests the Series 2 debug ports by prompting the user to put a 50 ohm resistor on each port one-at-a-time.
EFTMuxTestPath12	This action test the Series 2 instrument mux relays for ABus1 and ABus2.
EFTMuxTestPath34	This action test the Series 2 instrument mux relays for ABus3 and ABus4.
fetchDeviceByName	Diagnostics Action: This action returns the module handle of the module referenced by string name.
getICA	This action returns: 0 - no ICA defined 1 - Mass Interconnect (MAC panel) ICA 2 - Test System Interface (Express connect) ICA -1 - error condition
getModByName	This action returns the module handle of the module referenced by string name.
getModuleName	This action returns the string name of a module.
isThereA33120	This action returns a value of 1 if there is a 33120 Arbitrary Waveform Generator in the system.
isThereA34401	This action returns a value of 1 if a 34401 DMM is in the system.
isThereA53131	This action returns a value of 1 if a 53131 Counter is in the system.
PinAuxRelayTests	Tests the "Aux" relays for opens and shorts on the Pin Matrix Module.
PinAuxRelayTestsAll	Tests the "Aux" relays for opens and shorts on all of the Pin Matrix Modules in the system.
PinProtBypass	Closes relays to short the 200 ohm column protection resistors on the Pin Matrix Module.
PinRelayOpenTest	Tests the Pin Matrix Module for shorted relays.

Table 5-4. dgn (Diagnostic) Actions

PinRelayOpenTestAll	Tests all of the Pin Matrix Modules for stuck open relays.
PinRelayPathTest	Tests the Pin Matrix Module for stuck open relays.
PinRelayPathTestAll	Tests all the Pin Matrix Modules for stuck open relays.
PinReset	Reset Pin Matrix Module relays.
PinShortsTest	Test for shorts between pins on the Pin Matrix Module.
PinShortsTestAll	Test for shorts between pins on all of the Pin Matrix Modules in the system.
VI Debug Port Test	Tests debug ports 1-4 of the MCM using the DMM.
VI HV DC Accy	Tests HV DC source accuracy of the MCM using the DMM.
VI LV DC Accy	Tests LV DC source accuracy of the MCM using the DMM.
VIRelayTest1	Tests bus 1, 2 and UUTCOM matrix relays for opens and shorts in the MCM.
VIRelayTest2	Tests bus 3 and 4 matrix relays for opens and shorts in the MCM.
whichDmm	This action returns the system DMM as follows: 0 - No system DMM found 1 - E1411 with MCM 2 - 34401 with E8792 mux

Table 5-5. digitizer Actions

digitalReadRegCC	Low-Level Action: Reads a value from the specified E8794A custom card digital input register.
digitalWriteRegCC	Low-Level Action: Write a value to the specified Agilent E8794A Custom Card digital output register.
digitalWriteRegQueryCC	Low-Level Action: Returns the last value written to the specified register on the Agilent Technologies E8794A custom card .
Low-Level Actions	
ADCClear	Clears output buffer on digitizer (ADC) and aborts reading, if in progress.
ADCConfArm	Configures the arm subsystem of the E1563, E1564, or E1429 Digitizers.
ADCConfFreq	Configures the frequency subsystem of the E1563, E1564, or E1429 Digitizer (ADC).
ADCConfInControls	Configures the input circuit of the E1563, E1564, or E1429 Digitizer (ADC).
ADCFastOff	This action shuts off the Agilent1429 Digitizer's fast fetch mode until it is re-opened.
ADCGetSampleFreq	Returns the sample rate values from the most recent setup of the Agilent E1429 Digitizer.
ADCGetSampleInfo	Returns sample rate and data point count values from the most recent setup of the Agilent E1429 Digitizer (ADC).
ADCInitiate	Arms the E1563, E1564, or E1429 Digitizer (ADC) and retrieves the data after triggering occurs.
ADCIsSet	Waits until the E1563, E1564, or E1429 Digitizer (ADC) is ready for measurement.

Table 5-5. digitizer Actions

ADCLastReading	Transfers the current digitizer data from the E1563, E1564, or E1429 from memory into a waveform data type to allow easy viewing.
ADCReset	Resets the digitizer (ADC) to its power-on state for all channels.
ADCSelfTest	Runs a selftest on the Agilent E1429 Digitizer (ADC) that is provided by its Plug and Play driver.
ADCSetReturnMode	This action allows time-outs on the specified E1563, E1564 or E1429 Digitizer (ADC) to raise an exception when encountered.
ADCSetTimeout	Sets the E1563, E1564 or E1429 Digitizer (ADC) operation time out value (in milliseconds).
ADCSetTimeoutException	This action allows enabling/disabling exception response to a timeout on the specified E1563, E1564 or E1429 digitizer (ADC).
ADCStart	Arms the Agilent E1429 Digitizer (ADC) for the configured trigger event and returns.
ADCType	Returns the type of digitizer (ADC) installed: either E1429, E1563, or E1564.
digitalReadCC	Reads a value from the custom card digital input used in the Agilent E6198 switch/load unit.
digitalReadRegCC	Reads a value from the specified E8794A custom card digital input register.
digitalReadSU	Reads a value from the Agilent E6198 switch/load unit digital input ports.
digitalResetCC	Perform a soft reset on the Agilent E8794 custom card.
digitalWriteCC	Write a value to the Custom Card open collector digital output port used in the Agilent E6198 switch/load unit.
DigitalWriteQuery	Returns the current output state of the TS-5430 digital output ports.
digitalWriteQueryCC	Returns the current output state of the custom card open collector digital output port used in the Agilent E6198 switch/load unit.
digitalWriteQuerySU	Returns the current output state of the Agilent E6198 switch/load unit digital output ports.
digitalWriteRegCC	Write a value to the specified Agilent E8794A Custom Card digital output register.
digitalWriteRegQueryCC	Returns the last value written to the specified register on the Agilent Technologies E8794A custom card .
digitalWriteSU	Write a value to the Agilent E6198 switch/load unit digital output ports.
High-Level Actions	
ADC_Analyze_Wave	Measures a waveform using the E1563, E1564, or 1429 Digitizer to analyze it for Vmin, Vmax, High pulsewidth, Low pulsewidth and period values.
ADC_Config_1_Chan	Configures E1563, E1564, or E1429 for capturing a waveform for later analysis on Channel 1.
ADC_DCV	Returns instantaneous DC voltage from the Agilent Technologies E1563, E1564, or E1429 Digitizer using the "offset" parameter time from the trigger point.

Table 5-5. digitizer Actions

ADC_DCV-Avg	Returns average DC voltage from the Agilent E1429 Digitizer in a window defined by the "start" parameter and "stop" parameter time values from the trigger point.
ADC_Min_Max	Measures a waveform by the E1563, E1564, or E1429 Digitizer and directly analyzes it for Vmin & Vmax values.
ADC_Min_Max_(Win)	Returns Vmin or Vmax DC voltage values from the E1563, E1564 or E1429 Digitizer in a window defined by the "start" parameter and "stop" parameter times from the trigger point.
ADC_Transform	Transform E1563, E1564, or E1429 Digitizer returned data with Measurement Control Module attenuator gain and offset terms.

Table 5-6. DMM Actions

E1411/34401 DMM Low-Level Actions	
dmmConfCal	Configures the 50/60Hz line frequency and enables/disables autozero of the Agilent E1411 DMM.
dmmConfFunction	Sets the function, range and resolution of the Agilent E1411 or Agilent 34401 DMM.
dmmConfSample	Selects the sample source, sample count, and sample period of the Agilent E1411 DMM.
dmmConfSimple	Selects one of three measurement types on the Agilent E1411 or Agilent 34401 DMM for easy configuration.
dmmConfTrigIn	Selects the trigger input parameters of the Agilent E1411 or Agilent 34401 DMM.
dmmConfTrigOut	Enables you to route the Agilent E1411 DMM's "voltmeter complete" signal to the VXIbus TTL trigger lines.
dmmGetMultipleResults	Returns multiple readings from the DMM.
dmmGetResults	Returns the readings from the Agilent E1411 or Agilent 34401 DMM.
dmmInitiate	Initiates a reading cycle on the Agilent E1411 or Agilent 34401 DMM.
dmmIsSet	Waits until the Agilent E1411 or Agilent 34401 DMM is ready for measurement.
dmmReset	Resets the Agilent E1411 or Agilent 34401 DMM to its power-on state.
DmmSelfTest	Runs a selftest on the Agilent E1411 or Agilent 34401 Digital Multimeter (DMM) that is provided by its Plug&Play driver.
dmmSet	Sends current setup information to the Agilent E1411 or Agilent 34401 DMM.
dmmSetTimeout	Sets the Agilent E1411 orAgilent 34401 DMM operation time out value (in milliseconds).
dmmVerifyTerminalSwitch	Checks the state of the front/rear terminal switch on an Agilent 34401A DMM and compares the current switch state with the value in the parameter block.
E1411/34401 DMM High-Level Actions	
dmmMeasureACI	Measures AC current using the Agilent 34401 DMM, only.
dmmMeasureACV	Measures AC voltage using the Agilent E1411 or Agilent 34401 DMM.
dmmMeasureCurrent	Measures voltage across a user supplied shunt or sense resistor to calculate current using the Agilent E1411 or Agilent 34401 DMM.
dmmMeasureDCI	Measures DC Current using the Agilent 34401 DMM, only.
dmmMeasureDCV	Measures DC voltage using the Agilent E1411 or Agilent 34401 DMM.
dmmMeasureFrequency	Measures frequency using the Agilent 34401 DMM, only.
dmmMeasureOhms	Measures resistance using the Agilent E1411 or Agilent 34401 DMM.
dmmMeasurePeriod	Measures period using the Agilent 34401 DMM, only.

Table 5-6. DMM Actions

dmmMeasureTrigVoltage	Configures the Agilent E1411 or Agilent 34401 DMM to measure a DC voltage, triggered by a VXI backplane TTL signal (Agilent E1411 only) or External signal.
SM2040 Actions	
SM2040AnalogTrigRead	This function will arm the DMM, wait for an analog trigger, and return an array of measurements for the internal DMM buffer. Use the Abort button or a timeout value to prevent the measurement from waiting forever on a trigger. A return code of 99 indicates the abort condition and a return code of 88 indicates a timeout condition.
SM2040ArmAnalogTrigger	Setup the SM2040 for analog level trigger operation. Following reception of this command the DMM makes measurements continuously, waiting for a value which exceeds the threshold. Use SM2040ReadBuffer to get results.
SM2040ArmTrigger	Setup the SM2040 for external hardware trigger operation. Following reception of this command the DMM enters a wait state. Use SM2040ReadBuffer to get results.
SM2040BurstRead	This function is designed to read bursting measurements from the DMM, resulting from SM2040SetTrigRead and SM2040SetBurstRead operations. Set 'numReadings' to be number of settle samples + 1 + number of samples to take.
SM2040Cal	This function re-calibrates the DMM, and returns it to the current operating mode. This is an internal calibration.
SM2040ConfigureMeasurement	Sets the function, range, and rate to the selected values.
SM2040GetInfo	Returns the hardware and software versions as well as the type of card.
SM2040Read	SM2040Read reads the next result from the DMM. This routine can read all the Primary functions (those that can be selected using SM2040ConfigureMeasurement)
SM2040ReadBuffer	This function wait for a trigger, and return an array of measurements from the internal DMM buffer. Use the Abort button or a timeout value to prevent the measurement from waiting forever on a trigger. A return code of 99 indicates the abort condition and a return code of 88 indicates a timeout condition.
SM2040SetAutoRange	This function enables or disables autorange operation of the DMM.
SM2040SetBuffTrigRead	Setup the DMM for multiple triggered readings operation.
SM2040SetBurstRead	This function will set up the DMM to read bursting measurements.
SM2040SetFunction	Sets the DMM function to the selected value.
SM2040SetRange	Sets the DMM range to the selected value.
SM2040SetRate	Sets the DMM rate to the selected value.
SM2040SetRelative	Sets or clears the relative reading mode.
SM2040SetSynchronized	This function enables or disables the Synchronized operation of the DMM. Default operation is non-synchronized. Select the Synchronized mode when it is necessary to settle full scale input transitions from one reading to the next, and maintain the accuracy of the DMM. This is appropriate for VDC, Ohms, Leakage, DCI, Diode and Guarded Ohms. The result of the synchronized mode is a reduced measurement rate. To run synchronized, reading rate must be set to 10 rps or higher.
SM2040SetTrigRead	Setup the DMM for multiple triggered readings operation.
SM2040TrigRead	This function will arm the DMM, wait for a trigger, and return an array of measurements for the internal DMM buffer. Use the Abort button or a timeout value to prevent the measurement waiting forever for a trigger.

Table 5-7. dio (Digital I/O) Actions

Low-Level Actions	
ADDigitalRead	Reads digital output data from the specified digital port on the Advantech 1750 PCI device.
ADDigitalWrite	Writes digital output data to the specified digital port on the Advantech 1750 PCI device.
DigitalRead	Reads a value from the TS-5430 digital input ports.
DigitalWrite	Write a value to the TS-5430 digital output ports.
Spare Dig Read	Reads a value from the digital input port of the Agilent E1330 Digital I/O Module.
Spare Dig Write	Sends a value to the digital output port of the Agilent E1330 Digital I/O Module.

Table 5-8. event Actions

Low-Level Actions	
eventConf	Configures the Event Detector for a measurement.
eventInitiate	Starts the previously configured Event Detector measurement.
eventIsSet	Waits until the Event Detector is ready for operation.
eventReset	Resets the Event Detector to its power-on state.
eventSet	Sends current setup information to the Event Detector.
High-Level Actions	
eventGetResults	Returns stored data from the Event Detector.
eventMeasure	Makes a measurement using the Event Detector and stores data into memory.

Table 5-9. Generic Actions

Low-Level Actions	
DialogOkay	Display a message box with a prompt message and "Ok" button.
DialogOkayModal	Display a message box with a prompt message and "Ok" button.
DialogYesNo	Display a message in a dialog box; gives a Yes/No choice to continue in two different ways.
DialogYesNoModal	Display a message in a dialog box; gives a Yes/No choice to continue in two different ways.
GetFixtureID	Reads back the fixture I.
globalReset	Call all reset routines of instruments listed in the hardware configuration table.
msgDblIO	This action sends a string from an Instrument and receives a double response.

Table 5-9. Generic Actions

msgIntIO	This action sends a string from an Instrument and receives an integer response.
msgSend	This action sends a string to an instrument (INST).
msgStringIO	This action sends and reads strings from an instrument.
StoreLogString	This action allows the user to place any string into the log data storage area which will be used by any log record which accesses that field.
High-Level Actions	
ConcatenateStrings	The value of String2 is concatenated onto the end of String1.
DelayMicroseconds	Delays the test by the specified number of microseconds.
DelayMillisecond	Delays the test by the specified number of milliseconds.
DelaySecond	Delays the test by the specified number of seconds.
FormatString	The value of Argument is processed through the FormatString and will show up in ResultString.
IVIDriverDisable	Quickly places the instrument in a state where it has no, or minimal, effect on the external system to which it is connected. This state is not necessarily a known state.
IVIDriverErrorQuery	Queries the instrument and returns instrument specific error information. This function can be used when QueryInstrumentStatus is True to retrieve error details when the driver detects an instrument error.
IVIDriverLockObject	Obtains a multithread lock on the driver after waiting until all other execution threads have released their locks on the instrument session.
IVIDriverReset	Places the instrument in a known state and configures instrument options on which the IVI specific driver depends (for example, enabling/disabling headers). For an IEEE 488.2 instrument, Reset sends the command string *RST to the instrument.
IVIDriverSelfTest	Performs an instrument self test, waits for the instrument to complete the test, and queries the instrument for the results. If the instrument passes the test, TestResult is zero and TestMessage is 'Selftest passed'.
IVIDriverUnlockObject	Releases a previously obtained multithread lock.
LogDoubleToReport	This action allows the user to place any double into the log data file by producing a report record which will use the string argument.
LogIntToReport	This action allows the user to place any integer into the log data file by producing a report record which will use the string argument.
LogReport	This action allows the user to place any string into the log data file by producing a report record which will use the string argument.
LogReportImmediate.	This action allows the user to place any string into the log data file by producing a report record which will use the string argument.
NameSwitch	Switch string name.
ParameterLoad	Load (or write file if flag set) values for test parameters and skip control from[to] an ascii file.
UsecsElapsed	This action returns the time elapsed since the last "UsecsStart" Action.
UsecsStart	Starts clock for use with the "UsecsElapsed" Action.

Table 5-9. Generic Actions

UUTConnect	Connects to UUT pins by constructing Path names by adding a prefix (NamePrefix) to the base path name (BaseNames).
WaveformDataRead	Reads waveform sample data into waveform data type.
WaveformDataWrite	Write samples from 'hWaveform', one floating point number sample per line, to 'Filename.

Table 5-10. mcm Actions

Low-Level Actions	
viConfAttenuator	Configures the knee voltage of the MCM's programmable attenuator.
viConfCompare	Configures the MCM's comparator threshold voltage for triggering other instruments.
viConfSourceACV	Configures the MCM to source AC voltage.
viConfSourceDCI	Configures the MCM to source DC current.
viConfSourceDCV	Configures the MCM to source a DC voltage.
viIsSet	Waits until the MCM is ready for operation.
viReset	Resets the MCM to its power-on state.
viSet	Sends current setup information to the MCM.
High-Level Actions	
ForceIMeasV	Uses the MCM as a current source and measures the resultant voltage on the UUT using the DMM.
ForceIMeasValt	Uses the MCM as a current source to measures the resultant voltage on the UUT using the DMM.
ForceVMeasI	Uses the MCM as a voltage source and measures the resultant UUT current flow using the DMM.
viOutputACV	Configures the MCM to source AC voltages.
viOutputDCI	Configure the MCM to source DC current.
viOutputDCV	Configure the MCM to source DC voltage.
viSetCompare	Selects the comparator's trigger threshold voltage of the MCM.
viSetSourceACV	Configures the MCM to source AC voltages.
viSetSourceDCI	Configure the MCM to source DC current.
viSetSourceDCV	Configure the MCM to source DC voltage.

Table 5-11. power (Power Supply) Actions

Low-Level Actions	
psConfTrigIn	Selects the trigger input of the Power Supplies.
psConfVI	Selects output voltage and current of the Power Supplies.
psConnect	Enables output relays of the selected Power Supply.
psDisconnect	Disconnects output relays from the selected Power Supply.

Table 5-11. power (Power Supply) Actions

psGetCurrent	Reads and returns the output current of the Power Supply.
psGetStatus	Reads and returns Power Supply error status from last "psSet" or "psIsSet" Action.
psGetVoltage	Reads and returns the output voltage of the Power Supply.
psIsSet	Waits until the Power Supply is ready for output.
psProgVI	Programs and outputs the voltage/current of the Power Supply.
psReset	Resets the Power Supply to its power-on state.
psSelfTest	Runs a selftest on the Agilent Power Supply specified.
psSet	Sends configuration information to the Power Supply.
psSetReturnMode	Sets sequencer behavior on Power Supply timeout errors.
psSetTimeout	Sets Power Supply operation time out value (in milliseconds).
psSetVI	Selects the voltage and current parameters of the power supplies.
psTrigger	Send a software trigger to the Power Supply.

Table 5-12. serial Actions

Low-Level Actions	
scommBreak	Sends or clears a break signal.
scommFlush	Flushes the specified port.
scommGetConfig	Returns the current configuration of the given serial port.
scommReceive	Reads the specified number of characters from the Serial Port and checks those numbers against the "compareString" parameter.
scommReceiveBytes	Reads the Serial Port.
scommReset	Resets the Serial Port.
scommSend	Sends data to the Serial Port.
scommSendBytes	Sends user specified bytes of data to the Serial Port.
scommSet	Sends current setup information to the Serial Port.
scommSpawnSendBytes	Sends user specified bytes of data to the Serial Port.
scommStop	Flushes the transmit buffer used for the Serial Port and resets the number of bytes sent and received to 0.
scommTransmit	Writes to and reads from the Serial Port the number of bytes requested.
scommTransmitBytes	Writes to and reads from the Serial Port the number of bytes requested.
scommWaitForSendBytes	Sends user specified bytes of data to the Serial Port.

Table 5-13. switch Actions

Low-Level Actions	
GetLoadCardID	Reads the configuration register of the load box.
hardResetSU	Reset all Agilent E6198 switch units and all cards contained therein.
SUreset	Perform a soft reset on the Agilent Technologies E6198 switch unit.
High-Level Actions	
switching	This action connects / disconnects up to 10 switching paths.

Table 5-14. Load Card Actions

loadCardGetInfo	Returns information about the loADCard referenced by the instrument handle parameter.
loadCardReset	Routine to reset a given load card.
loadCardTest	Validates that a relay is set to a given position.

Table 5-15. SerialProtocol (Automotive Serial Protocol) Actions

CANGetInfo	Gets information on current state of J1939 or CAN interface including Bit Rate, Mode and Header Style (11 or 29 bit version).
CANRead	Reads CAN bus.
CANSetBitRate	Sets J1939 or CAN Interface bit rate in Bits/Sec.
CANSetIdMode	Sets Header Style (11 or 29 bit version).
CANSetReadFilter	Setup the CAN read filter.
CANSetTermination	Sets 124 ohm termination resistor across CAN bus (from CAN+ to CAN-).
CANSetUp	Performs general purpose set-up of CAN/J1939 protocol.
CANWrite	Writes to CAN bus.
CANWriteAlternate	Writes to CAN bus.
ISO9141EchoKeyword	Provides detection for 2 keywords and echos the complement keyword #2 as specified in ISO9141-2.
ISO9141FRead	Reads ISO9141 formatted data.
ISO9141FReadConfig	Constructs ISO9141 frame for the "ISO9141FRead" and "ISO9141Read" actions.
ISO9141FWrite	Writes formatted ISO9141 data.
ISO9141FWriteConfig	Configuration parameters for ISO9141 formatted write action.
ISO9141GetInfo	Gets information on the configuration and mode of operation of the ISO9141 serial interface device.
ISO9141InitSeq	Initializes the ISO9141 sequence.
ISO9141Read	Reads unformatted ISO9141 data.
ISO9141SetBitRate	Sets the bit rate of the ISO9141 serial protocol interface.
ISO9141SetByteTime	Sets the time delay between bytes in an ISO9141 message.
ISO9141SetUp	Provides general purpose set-up of ISO9141 protocol.

Table 5-15. SerialProtocol (Automotive Serial Protocol) Actions

ISO9141Write	Writes unformatted data to the ISO9141 protocol.
J1850BlockTransfer	Reads data from a file and sends the data along with header data to the J1850 device.
J1850GetConfig	Gets information on the configuration and mode of operation of the J1850 serial interface device.
J1850Read	Reads unformatted J1850 data.
J1850SetConfig	Sets information on the configuration and mode of operation of the J1850 serial interface device.
J1850SetSourceAddr	Sets the physical ID of the sender of the message.
J1850Write	Writes unformatted J1850 data.
J1939Read	Reads J1939 bus.
J1939Write	Writes unformatted J1939 data.
mComClrRxBuff	Clears the receive buffer of the associated serial protocol interface.
mComConfigGroup	Sets information on the configuration and mode of operation of the group message feature.
mComGetBoardInfo	Returns results of Multicom serial interface board self test, board serial number, board version number and firmware/hardware id numbers for on-board functions & Physical Interface Module (PIM) slots.
mComMask	Converts string to real number.
mComReset	Resets the Multicom box.
mComSetBitRate	Sets the bit rate of the associated serial protocol interface.
mComStartGroup	
mComStopGroup	Stops any group messages that were previously started by mComStartGroup.

Notes:

Index

B

C

D

O

P

Q

R

S

T